计算机仿真技术

Computer Simulation Technologies

MATLAB

翟亮 主编
王亚 凌志刚 等副主编
王耀南 主审

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

华信经管创新·市场营销系列
国家精品课程教材

计算机仿真技术

瞿　亮　主编
王　亚　凌志刚　王　石　王文洁　副主编
王耀南　主审

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

随着计算机技术的不断发展，仿真技术的应用领域在不断扩大，越来越受到重视，而作为仿真工具的 MATLAB 是美国 MathWorks 公司推出的科学计算软件，是一种广泛应用于工程计算及数值分析领域的高级计算机仿真语言，目前，MATLAB 已经成为国际上最流行的科学与工程计算的软件工具。

本书基于 MATLAB2011 版，介绍仿真的基础概念和方法，仿真工具 MATLAB 的环境、语法、数学运算和绘图功能，模块化建模的原理及基本算法；从应用领域的角度，介绍图像处理的基本内容和应用，控制系统的基本理论及仿真，电力系统仿真的理论和应用。

全书内容详实，结构清晰，力求做到理论与实践紧密结合。本书可作为高校电气类专业的高年级本科生与研究生仿真或计算机辅助设计课程的教材和参考书，也可作为电气类专业的工程技术人员及计算机开发人员的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目(CIP)数据

计算机仿真技术/瞿亮主编. — 北京：电子工业出版社，2013.5

ISBN 978-7-121-20048-9

I. ①计… II. ①瞿… III. ①计算机仿真 IV. ①TP391.9

中国版本图书馆 CIP 数据核字(2013)第 062094 号

责任编辑：田宏峰　　特约编辑：蒲　玥

印　　刷：三河市鑫金马印装有限公司

装　　订：三河市鑫金马印装有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱　　邮编：100036

开　　本：787×1092　1/16　印张：14.75　字数：377 千字

印　　次：2013 年 5 月第 1 次印刷

印　　数：4000 册　　定价：49.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010)88258888。

前　言

随着计算机技术的发展，计算机仿真得到非常广泛的应用，已成为科学研究的一种重要手段。对于所有理工科学生，掌握一门仿真工具，对于验证理论、设计以及分析系统而言，是非常重要的一种学习方法。在实际科研中几乎所有复杂系统都需要运用仿真技术进行系统的评估、设计和分析。

MATLAB 作为当前国际最流行的面向工程与科学计算的高级语言，可轻易地实现 C 语言或 FORTRAN 语言几乎全部的功能，并设计出功能强大、界面优美、稳定可靠的高质量程序，编程效率和计算效率极高。MATLAB 环境下的 Simulink 是当前众多仿真软件中功能最强大、最优秀、最容易使用的一个用于系统建模、仿真和分析的动态仿真集成环境工具，在各个领域都得到了广泛的应用。

本书介绍计算机仿真的基本概念原理，系统建模的方法和仿真算法、MATLAB 的语法、数学运算和数据可视化功能、Simulink 动态建模、数字图像的处理、控制系统仿真、电力系统仿真的原理及应用，书中程序基于 MATLAB2011 版。

本书内容由浅入深，根据仿真技术近年来在电气类专业中的发展情况和当前最新版的 MATLAB 的使用情况，以及多年来在教学和科研上的经验，结合实例进行介绍。全书共分 8 章，第 1 章介绍仿真的基础知识，第 2 章介绍 MATLAB 的环境、语法、数学运算功能，第 3 章介绍 MATLAB 的绘图功能，第 4 章介绍图像处理的基本内容及应用，第 5 章介绍系统建模方法和仿真算法，第 6 章介绍动态可视化建模工具 Simulink，第 7 章介绍控制系统的基本理论及仿真应用，第 8 章介绍电力系统仿真的原理及应用。考虑到系统仿真是一门实验性很强的学科，书中附有大量的示范程序。

本书的参考教学学时为 32 学时。在学习本课程之前，学生应已修过计算机程序设计语言、自动控制原理、电力系统理论、数字图像处理等课程。

本书可作为高校电气类专业的高年级本科生与研究生计算机仿真和计算机辅助设计课程的教材和参考书。也可作为电气类专业的工程技术人员及计算机开发人员的参考书。

由于 MATLAB 是一个涵盖内容丰富，而且功能完善的软件，我们不可能将其所有的功能一一介绍给读者，但本书涉及的内容基本覆盖了常用的分析工具和分析方法，并向读者详细阐述了 MATLAB 中实现这些方法的步骤。

本书由国内自动控制领域权威专家王耀南教授审阅，并提出了许多宝贵意见，此外蔡明杰、关培源、向杜君、姜艳君、贺佳俊也参与了部分编写工作，在此表示衷心的感谢。

由于作者水平有限，错误之处请读者不吝赐教，可以通过 E-mail（quliang2001@yahoo.com.cn）与作者联系。

作　者

2013 年 4 月

目　　录

第 1 章　计算机仿真概述

仿真是 20 世纪 40 年代末伴随着计算机技术的发展而逐步形成的一类试验研究的新兴方法。最初，仿真主要应用于航空、航天、原子反应堆等少数领域。计算机技术和信息科学的迅猛发展，为仿真技术的应用提供了技术和物质基础，目前仿真已渗透到国民经济的多个领域，成为分析、研究各种系统，尤其是复杂系统的重要工具，它不仅用于工程领域，如机械、航空、航天、电力、冶金、化工、电子等方面，还广泛用于非工程领域，如交通管理、生产调度、库存控制、生态环境以及社会经济等方面。

1.1　仿真的概念和方法

首先来看两个例子。

（1）气囊弹射速度确定（1997 年，美国）。汽车安全气囊的弹射速度原来为 220 英里/小时，在加拿大一年统计：在 6 000 件事故中，救了 4 000 人，打死了 2 000 人；通过大量的仿真实验后，1997 年 12 月美国众议院通过，速度调整为 180 英里/小时。

（2）美国三种典型导弹研制过程仿真技术的作用，如表 1-1-1 所示。

表 1-1-1　导弹研制过程仿真技术的作用

导 弹 类 型	原计划发射数量	仿真后实发	节 省 导 弹	节省费用（万美元）
爱国者	141	101	40	8 000
罗兰特	224	129	95	4 200
尾　刺	185	114	71	2 500

在现实生活及工程应用中，大部分的试验对象是很复杂的，并且要考虑安全性、经济性以及进行实验研究的可能性等，这在现场实验中往往不易做到，甚至根本不允许这样做。例如，在研究导弹飞行、宇航、反应堆控制等系统时，不经模拟仿真实验就进行盲目实验，将对人类的生命和健康带来很大的危险，这时，就需要对实际系统构建物理模型或数学模型进行研究，然后把对模型实验研究的结果应用到实际系统中，也就是仿真研究。

1.1.1　仿真的概念及分类

1．仿真的概念

“仿真”一词译自英文 Simulation，从字面上解释，表示“模拟真实世界”的意思。虽然人们很早就采用了利用模型来分析与研究真实世界的方法，但严格地讲，只有在 20 世纪 40 年代末，计算机（模拟计算机及数字计算机）的问世，才为建立模型及对模型进行试验提供了强有力的支持，仿真技术才获得了迅速的发展并逐步成为一门独立的学科。

仿真是以相似性原理、控制论、信息技术及相关领域的有关知识为基础，以计算机和各种专用物理设备为工具，借助系统模型对真实系统进行试验研究的一门综合性技术。它利用物理或数学方法来建立模型，类比模拟现实过程或者建立假想系统。以寻求过程的规律，研究系统的动态特性，从而达到认识和改造实际系统的目的。

系统仿真涉及相似论、控制论、计算机科学、系统工程理论、数值计算、概率论、数理统计、时间序列分析等多个学科。

相似性原理是仿真主要的理论依据。所谓相似，是指各类事物或对象间存在的某些共性，包括几何相似、性能相似、环境相似。例如，电路中的 RLC 振荡和机械中的弹簧振动都可以用相同的微分方程描述。

2．仿真分类

根据模型的属性，系统仿真分为物理仿真、数学仿真、半实物仿真。

1）*物理仿真*

按照真实系统的物理性质构造系统的物理模型，并在物理模型上进行实验的过程称为物理仿真。

物理仿真优点是：直观、形象，也称为“模拟”；缺点是：模型改变困难，实验限制多，投资较大。

2）*数学仿真*

对实际系统进行抽象，并将其特性用数学关系加以描述而得到系统的数学模型，对数学模型进行实验的过程称为数学仿真。计算机技术的发展为数学仿真创造了环境，数学仿真也称为计算机仿真。

数学仿真优点是：方便、灵活、经济；缺点是：受限于系统建模技术，即系统数学模型不易建立。

3）*混合仿真*

将系统的物理模型和数学模型以及部分实物有机地组合在一起进行实验研究，称为混合仿真仿真，也称为半实物仿真。

这种方法结合了物理仿真和数学仿真各自的特点，常常被用于特定的场合及环境中。例如，汽车发动机试验、家电产品的研制开发、雷达天线的跟踪、火炮射击瞄准系统等都可采用半实物仿真。

实际上，在工程实践中，以上各种仿真往往用于工程中的不同阶段。在工程设计分析阶段采用数学仿真，易于更改设计，具有灵活性和经济性；在部件子系统研制阶段，采用半实物仿真以提高仿真可信度和测试部件或子系统的功能；在最后定型阶段为了验证全系统的功能特性，则需要进行全物理仿真。

按照仿真系统与实际系统时间尺度上的关系，又可将其分为如下几类。

（1）实时仿真：仿真时钟与系统实际时钟完全一致，许多仿真应用需要满足实时性，这时往往需要实时操作系统或者专用实时仿真硬件的支持。

（2）欠实时仿真：仿真时钟比实际时钟慢，当对仿真的实时性没有严格的要求时，仿真时钟比实际时钟慢，不影响仿真的目的，采取欠实时仿真则可节约很多资金。

（3）超实时仿真：仿真时钟比实际时钟快，当实际系统周期太长时，若采用实际时钟就变得毫无意义，这时就要进行超实时仿真。对于大的、复杂的系统进行超实时仿真对计算机的计算速度要求是非常高的，如天气预报系统就需要超级计算机的支持。

1.1.2 仿真的基本步骤

仿真步骤如图 1-1-1 所示，具体如下所述。

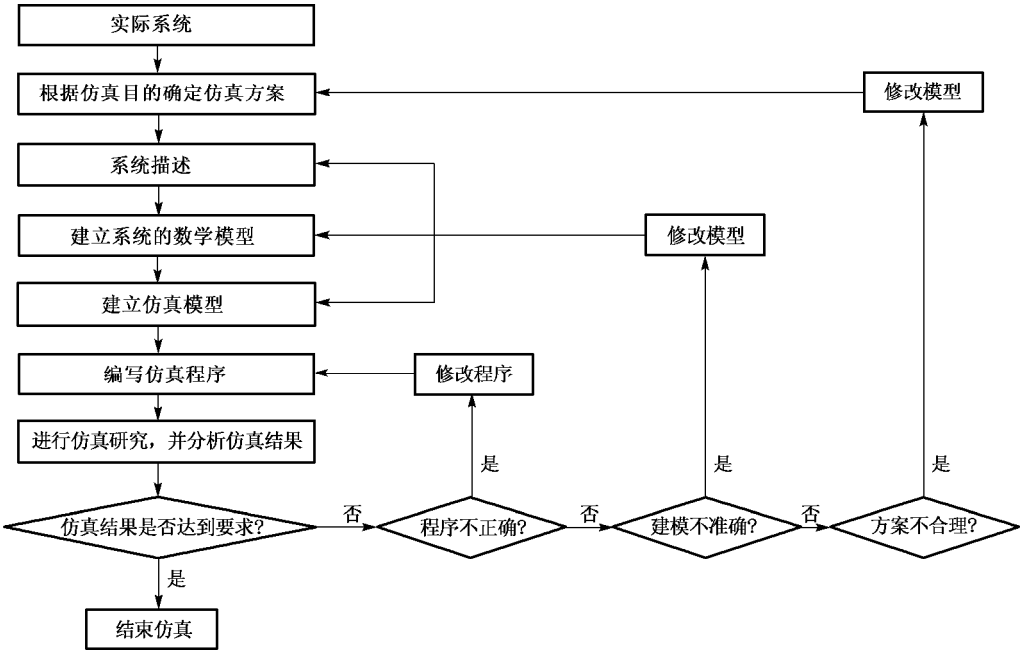


图 1-1-1　仿真步骤

1．根据仿真目的确定仿真方案

根据仿真目的确定相应的仿真结构和方法，规定仿真的边界条件与约束条件。

2．建立系统的数学模型

系统的数学模型是系统仿真的主要依据，是描述系统输入/输出变量以及内部各变量之间关系的数学表达式，描述系统各变量间静态关系采用静态模型，描述系统各变量间动态关系采用动态模型。例如，在控制系统中最常用的基本数学模型是微分方程与差分方程。

通常，根据系统的实际结构与系统各变量之间所遵循的物理、化学基本定律，如牛顿运动定律、基尔霍夫定律、动力学定律、焦耳-楞次定律等来列写变量间的数学表达式以建立系统的数学模型，这就是所谓用解析法来建立数学模型。

对于大多数复杂的系统，则必须通过实验的方法，利用系统辨识技术，考虑计算所要求的精度，略去一些次要因素，使模型既能准确地反映系统的动态本质，又能简化分析计算的工作，这就是所谓用实验法建立数学模型。

3．建立仿真模型

一般的数学模型都不能直接编制程序并用计算机求解，通常必须把数学模型转换成适宜编程并能在计算机上运行的模型——仿真模型。也就是需要通过一定的算法对原系统的数学模型进行离散化处理，就连续系统而言，就是建立相应的差分方程后再由计算机进行求解。

4．编制仿真程序

对于非实时仿真，可用高级语言依据相应的算法编程，而对于实时仿真往往采用汇编语言与高级语言共用的方式进行编程。

5．程序调试、验证模型、实验结果分析并确定实验方案

1）调试程序

调试程序的首要任务是检查并纠正程序的错误，使其在计算机上运行通过，并保证程序处于正确的工作状态。

2）验证模型

通过运行程序，用仿真实验数据与实际系统运行观测的数据结果相比较的方法，检验、确认模型能代表实际系统，可反映实际系统运行过程的特性。

3）根据实验结果的分析，确定实验方案

通过选择合理的参数实验范围，安排用较少的实验次数来达到预期的效果。如果结果不符合原有的设计要求，就应寻找原因并通过修改程序或修改仿真模型，反复多次运行程序直至达到设计要求。

应当注意，仿真研究是一个动态迭代过程，需要通过迭代过程逐步获取系统特性的信息。

1.2 仿真技术的应用和发展

1.2.1 仿真技术的应用

仿真技术的主要用途有

（1）优化系统设计。在实际系统建立以前，通过改变仿真模型结构和调整系统参数来优化系统设计。例如，控制系统、数字信号处理系统的设计经常要靠仿真来优化系统性能。

（2）系统故障再现，发现故障原因。实际系统故障的再现必然会带来某种危害性，这样做是不安全的和不经济的，利用仿真来再现系统故障则是安全的和经济的。

（3）验证系统设计的正确性。

（4）对系统及其子系统进行性能评价和分析，多为物理仿真，如飞机的疲劳试验。

此外，仿真可以作为教学设备来增强解析求解方法学的能力，用于训练的仿真模型使得学习成为可能，无须费用及现场指导。

目前仿真技术已应用到军事、生产制造等工程领域的各个方面。

1．军事领域

1）武器装备研制

仿真技术在武器装备研制过程中，使得在新武器研制计划开始前，能够充分利用仿真系统检验武器系统的设计方案和战术、技术性能的合理性，避免在实际研制过程中出现方案不合理的现象，从而缩短研制周期，并且可以支持技术评估、系统更新、样机研制，使得能够以较低的代价提高武器装备的战术性能。各用户（包括武器装备的研制部门、采购部门、训练部门和军事使用部门）可在合成环境中按需要综合应用各种仿真手段进行演习、训练和试验，鉴定现有的和研制中的武器装备的性能、战术部署和后勤保障。现在，在武器装备研制生产过程中，已规定将仿真系统列为必需的装备。

2）军事训练

分布式仿真系统通过联网技术将分散在各地的人，在回路中的仿真器、计算机生成的兵力，以及其他设备连接为一个整体，形成一个可以在时间和空间上互相耦合的虚拟战场合成环境，参与者可以自由地交互作用。这样，使过去主要依靠野战演习完成的任务可以利用计算机、仿真器和人工合成的虚拟环境来进行。技术的进一步发展还将把野外演习的部队和这种仿真器联系起来进行演习。利用仿真器产生动态的、直观的环境，配合仿真的地形、烟雾和“敌人”的武器装备，使部队能够进行生动逼真的军事演习。

3）先进概念与军事需求分析

在先进概念与军事需求分析方面（如使用新概念与先进技术的试验），对于未来军事行动中在条令、训练、指挥人员培养、组织、装备和士兵发展等方面的需求上，可以通过仿真和使用真实部队的士兵体验来评估技术综合集成的影响。

2．工业领域

同军事领域的需求和推动一样，由于工业系统的复杂性、大型化，出于安全性和经济性的考虑，仿真技术广泛应用于工业领域的各个部门。在大型复杂工程系统（项目）建设之前的概念研究与系统的需求分析过程中，仿真都发挥着越来越重要的作用。在电力工业中，随着单元发电机组容量的越来越大，系统变得越来越复杂，对它的经济运行、安全生产提出了更高的要求，仿真系统是实现这个目的的最佳途径。通过仿真系统可以优化运行过程，培训操作人员。电站仿真系统已成为电站建设与运行中必须配套的装备。核电站的运行必须安全，操作人员的技术素质、技能是保证安全运行的前提，仿真培训系统是提高操作人员素质、技能的有效手段。

在经济全球化、贸易自由化和社会信息化的今天，在技术更新速度加快的新形势下，制造业的经营战略发生了很大的变化，如何在最短的时间内，以最经济的手段开发出用户能够接受的产品，已成为今天市场竞争的焦点。虚拟制造是解决这个焦点问题的有效技术途径之一，它采用建模技术，在计算机及高速网络支持下，在计算机群组协同工作下，通过三维模型及动画实现产品设计、工艺规划、加工制造、性能分析、质量检验以及企业各级过程的管理与控制的仿真产品制造过程。虚拟制造是对已有的或未来的制造活动进行的

仿真过程，所进行的过程是仿真的，所生产的产品也是仿真的。仿真技术将在制造企业中发挥更加重要的作用。

3．其他应用领域

在为武器系统研制、作战训练和工业过程服务的同时，仿真技术的应用正不断地向交通、教育、通信、社会、经济、娱乐等多个领域扩展。近年来，国内研制了能够表述交通流特征和交通流质量的交通仿真软件平台，可以对交通规划、交通控制设计、交通工程建设方案等进行预评估。例如，在引黄入晋输水工程中，建立了全系统运行仿真系统。利用仿真系统验证了工程设计，提出了现有工程设计中影响运行的重大问题，寻找调度运行最佳模式等。

在医学仿真方面，建立了有关人体的生物学模型和三维视觉模型，为深入开展人体生命机理研究和远程医疗工作提供了有力的工具。

为了满足大容量、高速度通信网络研究的需要，对通信仿真的方法和软件开展了广泛的研究，为提高通信网络的性能和网络方案的优化提供了重要的分析和验证工具。

此外，仿真技术和虚拟现实技术在娱乐业中也显示出广阔的发展前景。

1.2.2 仿真技术的发展阶段

仿真技术的发展与控制工程、系统工程、计算机技术的发展密切相关。控制工程是计算机仿真技术较早的应用领域，其发展为系统仿真技术的形成和发展奠定了良好的基础。系统工程完善了建模与仿真的理论体系，使计算机仿真技术应用于非工程系统。计算机技术为计算机仿真的应用提供了强有力的工具和手段。

仿真研究的许多活动都是通过仿真软件来实现的，仿真软件是一类面向仿真用途的专用软件，其特点是面向问题、面向用户。近 40 年来，仿真软件充分吸收了仿真方法学、计算机、网络、图形/图像、多媒体、软件工程、自动控制、人工智能等技术所取得的新成果，从而得到了很大的发展。自 1955 年第一个仿真软件问世以来，按照新技术出现的时间顺序，可将仿真软件的发展分为以下六个阶段：

- 通用程序设计语言；
- 仿真程序包及初级仿真语言；
- 完善的商品化的高级仿真语言；
- 一体化（局部智能化）建模与仿真环境；
- 智能化建模与仿真环境；
- 支持分布交互仿真的综合仿真环境。

1.2.3 仿真技术的发展趋势

由于仿真理论、方法的提高，仿真试验任务的扩大以及相关学科的发展，展望仿真技术今后的发展趋势主要有以下几个方向。

1．向广阔的时空发展

以现代复杂军事系统为例，它涉及战略、战术、技术决策系统，指挥、通信、运输系统，外层空间、内层空间、武器和运载系统，地面与空间各军兵种、我友协同作战系统与作战环境等。这种激烈对抗的军事系统，对时空一致、任务协同、实时性、实用性等的要求都很高，因而在这类复杂仿真系统中有很多复杂、艰巨的技术问题亟待解决。

2．向快速、高效与海量信息通道发展

对于大型复杂系统、分布系统、综合系统进行实时仿真，由于信息量庞大，必须对信息进行快速、高效传输、变换和处理。以多微处理机为基础的全数字并行仿真计算机系统将会有更多的发展。

3．向规范化模型校核、验证、确认技术发展

模型建立后，如果没有规范化模型校核、验证、确认来检验、评价模型的正确性和置信度，仿真的精度和可靠性是无法保证的。目前它已引起仿真界的高度重视。

4．向虚拟现实技术发展

虚拟现实是将真实环境、模型化物理环境、用户融为一体，为用户提供视觉、听觉、嗅觉和触觉感官以逼真感觉信息的仿真系统，使人感到如同身临其境的仿真环境中。

5．向高水平的一体化、智能化仿真环境发展

开展系统仿真科学研究，开发仿真系统技术，需要一体化、智能化仿真环境等有效的工具，由于 PC 的大量推广和应用，将会发展适合这类机型的仿真软件。

6．向广阔的应用领域扩展与其他有关的学科融合

由于仿真的对象越来越广阔和复杂，即应用领域越广泛，相关的学科就越多，而且日趋密切，特别是在非工程或混合工程系统中，仿真技术的应用将会迅速增长。

随着仿真技术的发展，仿真技术应用目的趋于多样化、全面化。最初仿真技术是作为对实际系统进行试验的辅助工具而应用的，而后又用于训练目的，现在仿真系统的应用包括系统概念研究、系统的可行性研究、系统的分析与设计、系统开发、系统测试与评估、系统操作人员的培训、系统预测、系统的使用与维护等各个方面，它的应用领域已经发展到军用以及与国民经济相关的多个重要领域。

1.3 仿真工具 MATLAB

1.3.1 MATLAB 的发展历史

MATLAB 名字由 Matrix 和 Laboratory 两词的前三个字母组合而成，意为“矩阵实验室”。20 世纪 70 年代后期，时任美国新墨西哥大学计算机科学系主任的 Cleve Moler 教

授出于减轻学生编程负担的动机，为学生设计了一组调用 LINPACK 和 EISPACK 库程序的“通俗易用”的接口，此即用 FORTRAN 语言编写的萌芽状态的 MATLAB。

1983 年春天，工程师 John Little 与 Moler、Steve Bangert 一起开发了第二代专业版 MATLAB。1984 年，MathWorks 公司成立，MATLAB 正式推向市场。

目前，MATLAB 已经成为国际上最流行的科学与工程计算的软件工具，现在的 MATLAB 已经不仅仅是一个“矩阵实验室”，它已经成为了一种具有广泛应用前景的全新的计算机高级编程语言，有人称它为“第四代”计算机语言，在科学运算、自动控制与科学绘图领域，MATLAB 语言长期保持了独一无二的地位。

1.3.2 MATLAB 的特点

MATLAB 的五大功能是：

- 数值计算功能（Numeric Function）；
- 符号计算功能（Symbolic Function）；
- 图形和可视化功能（Graphic Function）；
- 记事本功能（Notebook Function）；
- 可视化建模和仿真功能（Simulink Function）。

MATLAB 用法简易、可灵活运用、程式结构强又兼具延展性，以下为其主要优点。

（1）语言简洁紧凑，使用方便灵活。MATLAB 程序的书写格式自由，数据的输入/输出语句简洁，很短的代码就可以完成其他语言要经过大量代码才能完成的、很复杂的工作。

例如，一条语句“A=[1 2 3;4 5 6;7 8 9]”就实现了对 3×3 矩阵的输入。

（2）数值算法稳定可靠，库函数十分丰富。MATLAB 的一个最大特点是强大的数值计算能力，它提供了许多调用十分方便的数学计算的函数，不必考虑数值的稳定性。

例如：

```
e=eig (A)                    %求矩阵 A 的特征值
[L, U]=lu (A)                %求矩阵 A 的 LU 分解，
polyder (b)                  %求多项式的微分
```

（3）运算符丰富。MATLAB 是用 C 语言编写的，所以 MATLAB 提供了和 C 语言几乎一样多的丰富的运算符，而且还重载了一些运算符，给它们赋予了新的含义。

例如：

```
C=A*B                        %矩阵的乘法
B=C'                         %求矩阵 C 的共轭复转置
x=A\b                        %求 Ax=b 的最小二乘解
```

（4）MATLAB 既具有结构化的控制语句，如 if、for、while，又支持面向对象的程序设计。

（5）语法限制不严格，程序设计自由度大。

例如，在 MATLAB 里可以不用先定义或声明变量就可以使用它们。

（6）程序的可移植性好。MATLAB 程序几乎不用修改就可以移植到其他的机型和操作系统中运行。

（7）MATLAB 的图形功能强大，支持数据的可视化操作，可方便地显示程序的运行结果。

（8）具有强大的工具箱。MATLAB 包含两个部分——核心部分和各种可选的工具箱。核心部分有几百个核心内部函数，工具箱则是由各个领域的高水平专家编写的，所以用户不必编写该领域的基础程序就可以直接进行更高层次的研究。例如，控制领域可以使用的工具箱就有 Control System（控制工具箱）、System Identification（系统辨识工具箱）、Robust Control（鲁棒控制工具箱）、Optimization（最优化工具箱）等。

（9）源程序具有开放性，系统的可扩充能力强。除了内部函数外，所有的 MATLAB 核心文件和工具箱文件都提供了 MATLAB 源文件，用户可通过对源文件的修改生成自己所需要的工具箱。

（10）MATLAB 是解释执行语言。MATLAB 程序不用编译生成可执行文件就可以运行，解释执行时程序执行的速度较慢，效率比 C 等高级语言要低，而且无法脱离 MATLAB 环境运行 MATLAB 程序，这是 MATLAB 的缺点。但是 MATLAB 的编程效率远远高于一般的高级语言，这可以使研究者把大量的时间花费在对控制系统的算法研究上，而不是浪费在大量的代码上。

1.3.3 MATLAB 的工具箱

MATLAB 最重要的特征是它拥有解决特定应用问题的程序组，也就是工具箱（Toolbox），如符号处理工具箱、控制系统工具箱、神经网络工具箱、模糊逻辑工具箱、通信工具箱和数据采集工具箱等许多专用工具箱。总的来说，迄今所有的几十个工具箱大致可分为两类：功能型工具箱和领域型工具箱。功能型工具箱主要用来扩充 MATLAB 的符号计算功能、图形建模仿真功能、文字处理功能以及与硬件实时交互功能，能用于多种学科；而领域型工具箱是专业性很强的，如控制工具箱、金融工具箱等。

MATLAB 工具箱具有很强的专门知识要求，它是为设计人员在运用某一专门理论解决问题时所提供的有效快捷的工具。对于大多数用户来说，要想灵活高效地运用这些工具箱，通常都需要学习相应的专业知识。只有在掌握了其理论的基础上才能够明白工具箱中每一个函数的意义、所要达到的目的和所要解决的问题，才能够正确地使用它们。

此外，开放性也是 MATLAB 最重要和最受人欢迎的特点之一。除内部函数外，所有的 MATLAB 主包文件和各工具箱文件都是可读可改的源文件，因为工具箱实际上是由一组复杂的 MATLAB 函数（M 文件）组成的，它扩展了 MATLAB 的功能，用以解决特定的问题，因此用户可以通过对源文件进行修改和加入自己编写的文件去构建新的专用工具箱。

MATLAB/Simulink 的主要产品及其相互关系如图 1-3-1 所示。

对于用户而言，除了可以使用随 MATLAB 版本所附带的大量工具箱之外，MATLAB 还有其他很多工具箱，这其中很多工具箱是免费的，而且这些工具箱覆盖的专业更加广泛。读者要了解这方面内容，可以到 Mathsworks 公司的相关网页上去查找。

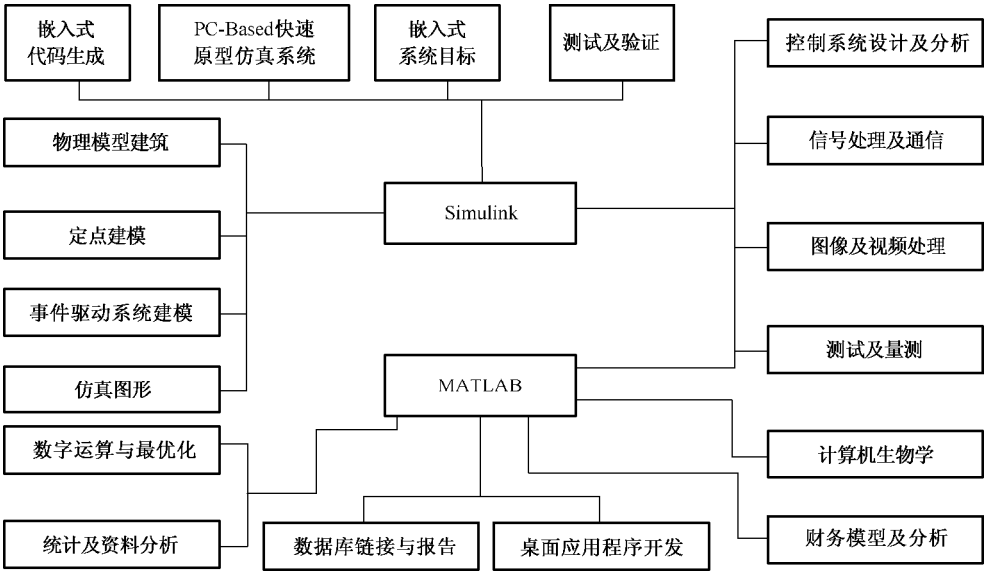


图 1-3-1　MATLAB/Simulink 的主要产品及其相互关系

1.3.4　Notebook

Notebook 是 MATLAB 将 Microsoft Word 和 MATLAB 集成在一起的一个工具，它不仅具备 Word 的全部功能，而且还具备 MATLAB 强大的数学处理能力和数据可视化能力。

Notebook 允许用户把在 Word 文档中创建的命令送到后台 MATLAB 中执行，然后将计算结果和绘制的图形送回到 Word，并插入到文档中。MATLAB Notebook 是“活”的笔记本，在该笔记本中的计算命令可随时修改、随时计算并生成图形。对于撰写科技报告、论文、专著的科技工作者，讲授、编写理工科讲义的教师，演算理工科习题的广大学生，MATLAB Notebook 都是非常有用的。

1．Notebook 的安装

首先安装 Word，然后启动 MATLAB，在其命令窗口输入“notebook-setup”此时，用户根据所用 Word 版本，在最后一行提示后面输入对应序号，并按回车键。MATLAB 会自动寻找 winword.exe 的安装路径，并在该路径下寻找模板文件 normal.dot。如果找到了，则出现提示“Notebook setup is complete”，表示 Notebook 安装结束。

启动 Notebook 有两种方法：从 Word 中启动或从 MATLAB 命令窗口直接输入“notebook”启动。

2．Notebook 的使用

安装启动后，Word 界面和通常的 Word 界面主要有两点区别：

（1）在菜单栏中多了一个 Notebook 菜单项，Notebook 的许多操作都可以通过该菜单项的命令来完成。

（2）在“文件”菜单项下多了一个 New M-book 命令项。如果在 M-book 模板下要建立新的 M-book 文档，可以选择该命令。

在 Word 中，通过菜单项对需要用 MATLAB 进行计算或绘图的语句进行操作，可直接在 Word 中看到执行结果、显示的格式及背景等参数也可通过菜单设置。

1.4 本书学习方法

计算机仿真技术涵盖的知识面很广，包括数学知识、专业知识和仿真语言。本书主要针对电气类专业，以 MATLAB 为仿真工具进行介绍。限于篇幅，涉及的专业知识和数学知识只做简要的介绍，详细内容读者可参考有关书籍，以求更深一步的了解。

MALAB 从根本上讲是一种函数型的语言。从语法的角度来看，MATLAB 通俗易懂，主要是学习大量函数及模块的应用。了解函数及模块的数学意义和包含的专业知识，熟练掌握函数和模块的使用方法，结合专业知识进行建模仿真，是学习仿真技术的主要内容。

MATLAB 中的函数有以下特点，读者在学习中应注意。

（1）函数数量很大，而且大多函数都牵涉到高等数学或相关专业的知识。

（2）绝大多数函数都有多种格式，表现为函数的参数和返回参数有多种变化形式，本书不可能一一列出，读者在使用中可以通过查看软件提供的帮助，找到自己需要的函数以及它们的各种用法。

（3）本书是基于 MATLAB2011 版编写的，不同版本的个别函数的使用方法会有所不同，读者可参考帮助说明。

计算机仿真是一门实验性很强的学科，读者应结合课堂理论学习内容，多进行上机操作。

第 2 章　数 学 运 算

数学建模及运算是仿真中最重要的一个环节，强大的数学运算功能是 MATLAB 的特色之一。MATLAB 包含了许多数学函数，从求和、正弦、余弦等基本函数到矩阵求逆、傅里叶变换等复杂函数，各领域的专家学者们开发的数值计算程序，使用了安全、成熟的算法，保证了最快的运算速度和可靠的结果。

2.1　MATLAB 的集成环境

启动 MATLAB 后，进入主窗口集成环境（见图 2-1-1），包括 MATLAB 主窗口、命令窗口（Command Window）、工作空间窗口（Workspace Window）、历史命令窗口（Command History Window）、工作空间浏览器（Workspace Browser）、路径浏览器（Current Directory Browser）、内存数组编辑器（Array Editor）、M 文件编辑/调试器（Editor/Debugger）、帮助窗口（Help）等。

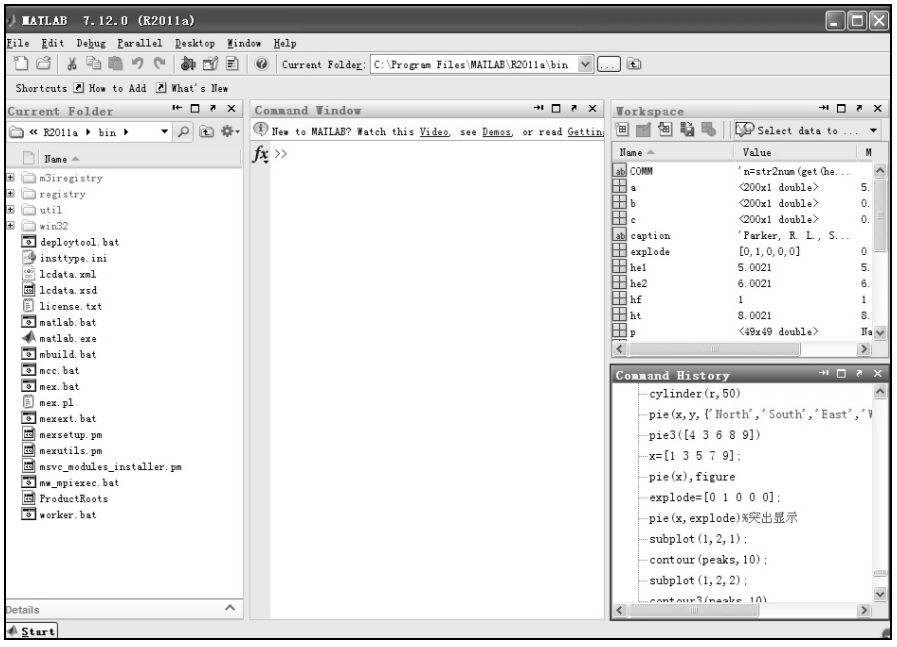


图 2-1-1　MATLAB 集成环境

1．主窗口

MATLAB 主窗口是 MATLAB 的主要工作界面。主窗口除了嵌入一些子窗口外，还包括菜单栏和工具栏等。

2．命令窗口

命令窗口是 MATLAB 的主要交互窗口，用于输入命令并显示除图形以外的所有执行结果。

3．历史命令窗口

历史命令窗口会自动保留自安装起所有用过的命令的历史记录，并且还标明了使用时间，方便用户查询。通过双击命令还可运行历史命令。如果要清除历史记录，可以选择 Edit 菜单中的 Clear Command History。

4．工作空间浏览器

工作空间窗口保存目前内存中所有的 MATLAB 变量的变量名、数据结构、字节数以及类型等信息，可对变量进行观察、编辑、保存和删除。

启动 MATLAB 后，会自动建立一个工作空间，工作空间在 MATLAB 运行期间一直存在，关闭 MATLAB 后工作空间自动消失。运行 MATLAB 程序时，程序中的变量被加入到工作空间中。除非用特殊的命令删除某变量，否则该变量在关闭 MATLAB 之前一直存在。某个时刻的工作空间中的所有变量可以保存到一个文件中，当下次启动 MATLAB 后，可用相关的命令把保存的工作空间的所有变量调入到当前工作空间。

关于工作空间变量管理的命令如下。

（1）who、whos：查看工作空间中的变量情况；

（2）clear：清除工作空间中的变量；

（3）size，length：求取变量的大小；

（4）exist：查询在当前的工作空间中是否存在一个变量；

（5）save：将工作空间中的变量保存到文件中；

（6）load：从文件中加载变量。

5．程序编辑/调试器

在程序编辑/调试器中可以建立、编辑、存储 M 文件，可以运行和调试（如断点、单步、跟踪、查看）程序。在命令编辑器中，不同的文本内容以不同的颜色显示。MATLAB 关键字为蓝色，注释语句为绿色，正在输入的字符串为红色，输入完毕的字符串为褐色，其他文本为黑色。文本的彩色显示便于程序的编辑和调试。

6．内存数组编辑器

利用内存数组编辑器可以输入大数组。首先，在命令窗口创建新变量；然后，在工作空间浏览器中修改变量的值，

7．路径浏览器

路径浏览器能够修改 MATLAB 的搜索路径并查看任一路径下的所有文件。只有在当前工作目录或搜索路径下的文件、函数可以被运行或调用。MATLAB 工作时根据 MATLAB 搜索路径，依次从各目录上搜索所需调用的文件、函数、数据。例如，当在命令控制窗口中输入一个字符 x 时，MATLAB 按以下顺序搜索：

（1）把 x 作为一个变量进行搜索，在当前工作空间中查找变量 x。

（2）把 x 作为一个内置函数进行搜索，查找函数 x 并执行。

（3）查找当前目录中的 M 文件 x.m。

（4）查找当前搜索路径中的 M 文件 x.m。

如果搜索路径中存在同名函数，则只会发现并执行路径中的第一个函数。

要查询某一命令是在搜寻路径的何处，可用 which 命令。

例如：

```
which solve
D:\MATLAB6p1\toolbox\symbolic\solve.m
```

一般来说，MATLAB 默认当前工作目录为 MATLAB\work。用户可在 MATLAB 桌面上的当前工作目录设定区进行修改。

8．帮助窗口

在 MATLAB 菜单中有帮助系统和系统演示 DEMO。由于 MATLAB 有大量的函数和工具箱，并且这些函数、工具箱随着软件版本的升级还在不断地扩充。对用户来说，借助于 MATLAB 自身的帮助及演示系统寻求函数、工具箱的使用方法，是掌握 MATLAB 的重要方法。

在命令窗口中输入 help 命令就可以在命令窗口显示相关内容的帮助，

```
help                  %打开 help 浏览器
help  函数名/文件名   %显示相关函数/文件的帮助体
lookfor  关键词       %查找所有的 MATLAB help 标题以及 MATLAB 搜索路径中的 M 文件的
                      %第一行，返回结果为包含所指定的关键词的项。
```

例如，查找包含积分这个关键词的所有命令。

```
lookfor integral
```

MATLAB 6.0 以上的版本提供了一种类似模糊查询的命令查询方法，用户只需要输入命令的前几个字母，然后按 Tab 键，系统就会列出所有以这几个字母开头的命令。

2.2 MATLAB 语法

MATLAB 语言从根本上讲是一种函数型的语言。相对目前其他高级语言而言，MATLAB 的语法简单易学，有关变量、运算符及流程控制语法大多与流行语言相似。

2.2.1 变量

1．命名规则

MATLAB 中所有变量的数据类型都是复数矩阵，因此在变量使用之前，不需要指定变量的数据类型，也不必事先声明变量。

在 MATLAB 中，所有的数据均以矩阵的形式存储，每个矩阵的单元可以是数值类型、逻辑类型、字符类型或者其他任何数据类型。

标量可以用 1×1 矩阵来表示；一组 *n* 个数据可以用 1×*n* 矩阵来表示；多维数组可以用多维矩阵来表示。可用命令 whos 来显示数据的类型、存储空间等信息。

MATLAB 变量的命名规则如下：

- 变量名必须是不包含有空格的单个词；
- 变量名区分字母大小写；
- 变量最多不超过 31 个字符，第 31 个字符之后的字符将被忽略；
- 变量名必须以字母打头，之后可以是任意字母、数字或下划线。

由于许多标点符号在 MATLAB 中具有特殊含义，所以变量名中不允许使用。

例如：

```
B=[1+9i,2+8i,3+7j; 4+6j 5+5i,6+4i; 7+3i,8+2j,i]
```

其中，矩阵中各行元素由分号分隔，而同行不同元素由逗号或空格分隔。

MATLAB 中数值的记述采用习惯的十进制表示法，对很大及很小的数可采用科学记数法，如 1.3e+18 表示 1.3 乘以 10 的 18 次方。所有的数都按双精度浮点方式保存。如需将计算结果以不同的精确度的数字格式显示，可以在命令视窗上的功能菜单上的 Options 下选 Numerical Format，或者直接在命令视窗输入以下的各个数字显示格式的命令，以 π 值为例，不同显示精度见表 2-2-1。

表 2-2-1　数字的显示精度

指　令	含　义	举　例
format short	小数点后 4 位有效数字，最多不超过 7 位。大于 1 000 时，用科学记数法表示	3.1416e+003
format short e	5 位科学记数表示	3.1416e+00
format long	15 位数字表示	3.14159265358979
format rat	近似有理数表示	355/133

2．数值类型

数值类型包括整数、浮点数、复数、inf、nan。inf 和-inf 分别表示正无穷大和负无穷大，除法运算中除数为 0 或者运算结果溢出都会导致 inf 或-inf 的运行结果。在 MATLAB 中用 nan（Not a Number）来表示一个既不是实数也不是复数的数值。

字符串在 MATLAB 中被看成一个行向量，由一对单引号给定，每个字符构成向量的元素。例如，str='hello'等价于 str=['h','e','l','l','o']。

系统默认变量在 MATLAB 启动时由系统自动生成，用户在编写命令和程序时，应避免使用如表 2-2-2 所示的系统变量。

3．矩阵的创建及元素操作

MATLAB 的基本运算单元就是矩阵，因此矩阵创建和矩阵元素操作是数学运算的基础。简单矩阵可直接插入数据元素，复杂矩阵的创建有很多方法。

表 2-2-2　系统默认变量与常数

变量与常数	含　义	变量与常数	含　义
ans	计算结果的变量名	computer	确定运行的计算机
eps	浮点相对精度	inf	无穷大
i	虚数单位	inputname	输入参数名
nan	非数	nargin	输入参数个数
nargout	输出参数的数目	pi	圆周率
nargoutchk	有效的输出参数数目	realmax	最大正浮点数
realmin	最小正浮点数	varargin	实际输入的参量
varargout	实际返回的参量	—	—

- 利用冒号表达式建立一个向量；
- 用特殊函数生成矩阵；
- 利用数组编辑器输入大数组；
- 大矩阵可由方括号中的小矩阵或向量聚合建立；
- 针对复杂的矩阵建立一个 M 文件，供以后使用；
- 利用外部数据文件装入到指定矩阵。

构造矩阵时，如果矩阵的数据类型不同，则 MATLAB 会自动对某些元素进行类型转换，然后生成的矩阵具有相同的类型。获取矩阵的元素也有多种方法，常用的有

- 用编号索引；
- 使用线性索引；
- 使用冒号；
- 使用 end 关键字。

例如，矩阵 ***A*** = [2 6 9; 4 2 8; 3 0 1]，若想得到矩阵 ***A*** 第 3 行第 2 列的元素值，可以使用编号索引 A (3,2)或使用线性索引 A (6)。

MATLAB 中冒号运算符具有很多功能，既可以创建矩阵，也可以访问矩阵的特定行、列或元素。

当冒号用于创建向量时：

（1）m:n 表示向量[*m*,*m*+1,*m*+2,…,*n*]，当 *m*>*n* 时，将产生一个空向量。

（2）m:k:n 表示向量[*m*,*m*+*k*,*m*+2*k*,…，*m*+*i***k*]，其中 *i*=fix(*n*–*m*)/*k*；fix 为取整函数。

例如，对矩阵的元素操作：

（1）提取矩阵 ***A*** 的第 *n* 行第 *m* 列的元素，表示为 A(n, m)。

（2）提取矩阵 ***A*** 的第 *n* 行的所有元素，表示为 A(n, :)。

（3）提取矩阵 ***A*** 的第 *m* 列的所有元素，表示为 A(:, m)。

（4）将矩阵 ***A*** 的第 *n* 行第 *m* 列的元素重新赋值 *b*，表示为 A(n, m)=b。

（5）将矩阵 ***A*** 的第 *n* 行的所有元素重新赋值 *b*，表示为 A(n, :)=b。

（6）将矩阵 ***A*** 的第 *m* 列的所有元素重新赋值 *b*，表示为 A(:, m)=b。

（7）将矩阵 ***A*** 的第 *n* 行第 *m* 列的元素删除，表示为 A(n, m)=[ ]。

（8）将矩阵 ***A*** 的第 *n* 行的所有元素删除，表示为 A(n, :)=[ ]。

（9）将矩阵 ***A*** 的第 *m* 列的所有元素删除，表示为 A(:, m)=[ ]。

（10）提取矩阵 ***A*** 第二行开始后的所有元素 A(2:end,:)

有很多创建特殊矩阵的函数，如表 2-2-3 所示。

表 2-2-3　创建特殊矩阵的函数

函　　数	功　　能
ones	创建一个所有元素都为 1 的矩阵
zeros	创建一个所有元素都为 0 的矩阵
eye	创建对角线元素为 1，其他元素为 0 的矩阵
accumarray	将输入矩阵的元素分配到输出矩阵中的指定位置
diag	根据矢量创建对角矩阵
magic	创建一个方形矩阵，其中行、列和对角线上元素的和相等
rand	创建一个矩阵，其中的元素为服从均匀分布的随机数
randn	创建一个矩阵，其中的元素为服从正态分布的随机数
randperm	创建一个矢量(1×*n* 的矩阵)

【例 2-2-1】产生一个在区间[10,20]内均匀分布的 4 阶随机矩阵。

```
a=10;b=20;
x=a+(b-a)*rand(4)
```

4．几个易混淆的概念

标量、向量、数组、矩阵、胞元数组是几个容易混淆的概念。

MATLAB 的基本运算单位是矩阵，通常矩阵与数组的意义相同，都是指含有 *m* 行与 *n* 列数字的矩形结构。两者在 MATLAB 中的运算性质不同，矩阵采用线性代数的运算方式，而数组强调元素对元素的运算（又称为矩阵的点运算）。

只有一行或一列的矩阵就是向量（或称为矢量），分别称为行向量及列向量。例如：

```
b=[1,4,8,2]                    %b 是一个行向量
c=[154;8;2]                    %c 是一个列向量
```

只有一个元素的矩阵就是标量（或称为常数），如 d＝[1]或 d＝1。

可以用函数 length(x)计算向量长度，如果 x 是一个常数，则 length(x)返回 1；如果 x 是一个向量，则 length(x)返回向量的长度。

此外，MATLAB 中还有一种类似 C 语言 struct 的数据类型，即胞元数组 cell，可将不同类型的数据保存在同一个变量中，例如：

```
b={'test', 2009};
b{1}
ans =
test
```

2.2.2 运算符

MATLAB 的运算符可以分为以下三大类。

（1）算术运算符：用来进行相关的数学运算，如加、减、乘、除、乘方等；

（2）关系起算符：进行数值或矩阵的大小比较；

（3）逻辑运算符：进行相关的逻辑运算，如与、或、非等。

操作符与特殊字符如表 2-2-4 所示。

表 2-2-4　操作符与特殊字符

操作符和特殊字符	含　义	操作符和特殊字符	含　义
+	加	-	减
*	矩阵乘法	.*	数组乘（对应元素相乘）
^	矩阵幂	.^	数组幂（各个元素求幂）
\	矩阵左除	/	矩阵右除
.\	数组左除	./	数组右除
:	冒号	( )	圆括
[ ]	方括	.	小数点
..	父目录	...	继续
,	逗号（分割多条命令）	;	分号（禁止结果显示）
%	注释	!	感叹号
'	转置或引用	=	赋值
==	相等	<>	不等于
&	逻辑与	|	逻辑或
~	逻辑非	xor	逻辑异或

在 MATLAB 的算术运算符中，要注意区分矩阵运算与数组运算。大多运算符为数组运算，即同维的数组对应元素进行运算或数组与标量的运算。而矩阵运算则需满足矩阵运算中维数匹配的基本条件。

1．算术运算符

MATLAB 的算术运算符可按优先级由低到高分为 5 级，每一级内优先级相同，运算时从左向右结合。各级所包含的运算符如下：

（1）转置符（.）、幂符（.^）、复共扼转置（'）、矩阵幂符（^）、数组幂符（.^）。

（2）标量加（+）、标量减（–）。

（3）向量乘法（.* ）、向量右除（./ ）、向量左除（.\ ）、矩阵乘法（*）、矩阵右除（/）、矩阵左除（\）。

（4）加法（+）、减法（–）。

（5）冒号运算符。

2．关系运算符

MATLAB 所有的关系运算和逻辑运算均按数组运算规则定义，MATLAB 提供了 6 种关系运算符，即<（小于）、<=（小于或等于）、>（大于）、>=（大于或等于）、==（等于）、～=（不等于）。

值得注意的是，关系运算符和逻辑运算符只针对两个相同长度的矩阵，或其中之一是标量的情况进行运算。对于前者，是指两个矩阵的对应元素进行比较，返回具有相同长度的矩阵；对于后者，是指这个标量与另一个矩阵的每个元素进行运算。运算结果只有 0 和 1 两种情况，0 表示不满足条件，1 表示满足条件。

【例 2-2-2】对矩阵 ***A*** 和 ***B*** 进行比较。

程序如下：

```
A=magic(3);                                    %矩阵 A 和 B
B=[1 2 3;4 5 6;7 8 9];
C=A>B                                          %比较 A,B 的大小,分别赋给变量 C 和 D:
C=
1 0 1
0 0 1
0 1 0
```

【例 2-2-3】建立矩阵 ***A***，然后找出大于 4 的元素的位置。

程序如下：

```
A=[4,-65,-54,0,6;56,0,67,-45,0]        %建立矩阵 A
find(A>4)                              %找出大于 4 的元素的位置
ans =
    2
    6
    9
```

关系运算符对于程序流程控制非常有用，在循环和条件控制语句中经常用到。

3．逻辑运算符

MATLAB 提供了 4 种逻辑运算符，即&（与）、|（或）、～（非）和 xor（异或）。逻辑运算符是用来处理两个运算元，在逻辑运算中，规定非零元素的逻辑量为“真”，用代码“1”表示；零元素的逻辑量为“假”，用代码“0”表示。

例如：

```
a=magic(3);b=ones(3,3)
a&b
ans=
1 1 1
1 1 1
1 1 1
~a
```

```
ans=
0 0 0
0 0 0
0 0 0
```

MATLAB 还提供了大量的逻辑函数，例如：

（1）异或函数 xor（A,B）：A 或 B 非零返回 l，A 和 B 都是零或都是非零返回 0。

（2）any（A）：检测矩阵中是否有非零元素，如果有，则返回 1；否则，返回 0。

（3）all（A）：检测矩阵中是否全为非零元素，如果是，则返回 1。否则，返回 0。例如：

```
A=[0 1 2];any(A)
A=[1 0 3;2 0 1 ];any(A)
ans=
1
ans=
1 0 1
```

2.2.3 流程控制

MATLAB 的流程控制包括顺序结构、循环结构、条件结构等，语法格式与其他语言基本一致。

1．顺序结构

顺序结构就是由前至后依次执行程序的各条代码，直至最后一条代码。脚本文件就是典型的顺序结构。

2．循环结构

循环结构是按照给定的条件，重复执行指定的代码。该结构一般用于有规律的重复运算。在 MATLAB 中包括 for 循环和 while 循环。

1）for…end 语句

for 语句使用灵活，通常用于循环次数已经确定的情况。其调用格式为：

```
for 变量名=表达式
    循环体语句组
end
```

其中，表达式一般是以“s1:s2:s3”的形式给出。s1 表示循环的初始值，s2 表示步长（若 s2 省略，则表示步长为 1）。对于正的步长，当变量大于 s3 时结束循环；对于负的步长，当变量小于 s3 时结束循环。

2）while 语句

while 语句一般用于实现不能确定循环次数的情况。只要条件表达式成立，循环体语句就会一直被执行。while 语句的基本形式是：

```
while 条件表达式
      循环体语句组
end
```

【例 2-2-4】找出[100,1000]以内的全部素数。

程序如下：

```
n=0;
for m=100:1000
   flag=1; j=m-1;
   i=2;
   while i<=j & flag
       if rem(m,i)==0            %rem 为除后余数
          flag=0;
        end
       i=i+1;
   end
   if flag
      n=n+1;
      prime(n)=m;
    end
end
prime                            %变量 prime 存放素数
```

【例 2-2-5】从键盘输入若干个数，当输入 0 时结束输入，求这些数的平均值及其和。

程序如下：

```
sum=0;
cnt=0;
val=input('Enter a number (end in 0):');
while (val~=0)
     sum=sum+val;
     cnt=cnt+1;
     val=input('Enter a number (end in 0):');
end
if (cnt >0)
    sum
    mean=sum/cnt
end
```

3．选择结构

选择结构根据给定的条件来执行不同的代码。在 MATLAB 中有 if-else-end 和 switch-case-otherwise 两种结构。

```
if (逻辑运算式…)
    （程序组）
```

```
else
    （程序组…）
elseif
    （程序组）
end
```

switch 开关表达式为：

```
case 表达式 1
    语句组 1
case {表达式 1,表达式 2,…,表达式}
    语句组 2
...
otherwise
    语句组 n
end
```

【例 2-2-6】输入一个字符，若为大写字母，则输出其对应的小写字母；若为小写字母，则输出其对应的大写字母；若为数字字符则输出其对应的数值，若为其他字符则原样输出。

程序如下：

```
c=input('请输入一个字符','s');
if c>='A' & c<='Z'
    disp(setstr(abs(c)+abs('a')-abs('A')));
elseif c>='a'& c<='z'
    disp(setstr(abs(c)-abs('a')+abs('A')));
elseif c>='0'& c<='9'
    disp(abs(c)-abs('0'));
else
    disp(c);
end
```

【例 2-2-7】通过菜单方式选择颜色。

```
s=menu('colorselection','red','green','blue','yellow')
switch s
case 1,scolor='red';
case 2,scolor='green';
case 3,scolor='blue';
case 4,scolor='yellow';
otherwise disp('Error!')
end
```

4．试探式结构

试探式语句是 MATLAB 独有的结构，先试探性地执行语句组 1，如果在此段语句执行过程中发现错误，则将出错信息赋给系统保留变量 lasterr，并放弃执行这段语句，转而去执行语句组 2。其基本格式为：

```
try
    语句组 1
catch
    语句组 2
end
```

【例 2-2-8】先求两矩阵的乘积，若出错，则自动转去求两矩阵的点乘（矩阵乘法运算要求两矩阵的维数相容，否则会出错）。

程序如下：

```
A=[1,2,3;4,5,6]; B=[7,8,9;10,11,12];
try
   C=A*B;
catch
   C=A.*B;
end
C
lasterr          %显示出错原因
```

5．常用控制命令

break：用来终止当前循环；

continue：终止本次循环并继续下次循环；

return：终止本次函数调用或终止键盘输入的模式；

echo：执行 M 文件时，在命令窗口显示执行过程；

error：显示出错信息并终止程序；

input：用来提示并接收用户从键盘输入数据、字符串或表达式的值；

pause：用于暂时中止程序的运行，常用于程序的调试和查询中间变量值。

2.2.4 M 文件

在 MATLAB 中用户经常用到的文件类型只有两种，一种是代码组成的 M 文件，另一种是在后续章节中基于模块建模的框图文件，后缀名分别是.m 和.mdl。

M 文件分为 M 命令集及 M 函数。M 命令集的效用和将命令在命令窗口逐一输入完全一样，因此在 M 命令集中可以直接使用工作空间的变量，在 M 命令集中设定的变量，在工作空间中可见。M 函数则需要用到输入参数和输出参数来传递信息，并且函数中定义的变量属于局部变量，只在函数体中有效。如需定义为全局变量，需用 global 语句声明。

同 C 语言的函数一样，MATLAB 命令通常是用小写字母书写。

1．M 文件的创建及编辑

M 文件是一个文本文件，它可以用任何编辑程序来建立和编辑，常用 MATLAB 自己提供的文本编辑器。

【例 2-2-9】建立 M 命令集，要求根据输入的半径，计算圆的面积和球的体积。

程序如下：

```
r=input('Type radius:');
area=pi*r^2;
volume=(4/3)*pi*r^3;
fprintf('The radius is %12.4f\n',r)
fprintf('The area of a circle is %12.4f\n',area)
fprintf('The volume of a sphere is %12.4f\n',volume)
```

2．M 函数

M 文件除了可以撰写程序外，还可以用来定义函数。用户自定义函数可以和系统内建的函数（如 sin、cos 等）一样自由使用。

M 函数的格式为：

```
function [输出参数表]=函数名(输入参数表)
函数体
end
```

M 函数定义的语法有一些规定：

（1）函数文件名必须与函数名相同。

（2）函数的第一行必须以关键字 function 开始，表明该文件是函数文件。输入参数表是以逗号相分隔的形参。输出表是函数的返回值，如果返回值只有一个，方括号可以省略；当返回值不止一个时，输出参数表中的各个变量要以逗号隔开。这里输入和返回变量的实际个数分别由 nargin 和 nargout 两个 MATLAB 保留变量来给出，只要进入该函数，MATLAB 就将自动生成这两个变量，不论是否直接使用这两个变量。

（3）函数名称的取法的规定与一般变量相同。

（4）函数体是实现函数功能的部分，包括函数调用、程序流程控制、输入、输出、赋值、计算等语句。

（5）注释部分，用百分号%开头。

对于定义好了的函数，在命令输入窗口或其他文件中均可调用。在调用函数时，参数可以是常量、有确定值的变量或表达式。函数调用可以嵌套，一个函数可以调用别的函数，甚至调用它自己（递归调用）。

【例 2-2-10】计算 x 的阶乘。

程序如下：

```
function p=jc(x)
p=1;
for i=1:x
p=p*i;
end
p
end
```

此函数可在其他 M 文件或命令窗口调用，如在命令窗口下输入 jc(4)，在该窗口可看到：

```
p=
24
```

2.2.5 编程技巧

MATLAB 编程是程序开发的一种，应该符合一般程序开发的规律。良好的编程习惯可以提高工作效率，减少不必要的失误。对于初学者来说，应该注意以下几个方面：

（1）数据结构必须事先规划好，如果数据结构设计存在错误或不妥，那么程序修改的工作量将是巨大的。

（2）尽量避免使用全局变量。

（3）函数尽可能功能简明，使其可以重用，从而程序实现模块化。

（4）良好的编写风格，使得别人或者自己能够容易读懂之前所写的代码。具体的命名应具有较明确的意义；代码层次分明；注释清楚且充分等。

（5）注重程序的充分测试，注意警告信息。

（6）具有建立和求解数学模型的能力，能够简化程序的复杂性。

由于 MATLAB 语言是解释性语言，所以有时程序的执行速度不是很理想。在程序编写中可以注意以下事项，以便优化程序，加快程序的执行速度。

（1）及时清空工作空间。在每一个程序的开头加上 clear 命令，以清空内存中的自定义变量。

（2）直接使用矩阵与向量，避免使用循环。循环语句及循环体经常被认为是 MATLAB 编程的瓶颈问题。改进这样的状况有两种方法：首先尽量用向量化的运算来代替循环操作；其次，在必须使用多重循环的情况下，如果两个循环执行的次数不同，则建议在循环的外环执行循环次数少的，内环执行循环次数多的，这样也可以显著地提高速度。

（3）不显示不必要的中间结果。

（4）对大型矩阵预先定义维数。给大型矩阵动态地定维是个很费时间的事，建议在定义大矩阵时，首先用 MATLAB 的内在函数，如 zeros()或 ones()对矩阵先进行定维，然后再进行赋值处理，这样会显著减少所需的时间。

（5）优先使用内部函数。矩阵运算应该尽量采用 MATLAB 的内在函数，因为内在函数是由更底层的编程语言 C 构造的，其执行速度显然快于使用循环的矩阵运算。

（6）采用有效的算法。在实际应用中，解决同样的数学问题经常有各种各样的算法。例如，求解定积分的数值解法在 MATLAB 中就提供了两个函数 quad 和 quad8，其中后一个算法在精度、速度上都明显高于前一种方法。如果一个方法不能满足要求，可以尝试其他的方法。

（7）充分利用其他高级语言。如果采用很多优化措施，但执行速度仍然很慢，这就应该考虑用其他语言，如 C 或 Fortran 语言。按照 Mex 技术要求的格式编写相应部分的程序，然后通过编译链接，形成在 MATLAB 中可以直接调用的动态链接库（DLL）文件，这样可以显著地加快运算速度。

（8）测试程序执行时间。tic 命令用于启动秒表，toc 命令用于停止秒表，返回的是变量 elapsed_time。

【例 2-2-11】考虑生成一个 5×10 000 的 Hilbert 长方矩阵，该矩阵的定义是其第 i 行第 j 列元素为 $h\{i,j\}=1/(i+j-1)$。

以下用不同的方法进行计算，计算时间会不同。程序如下：

方法 1

```
tic                    %tic 与 toc 是用于记时的命令
for i=1:5
for j=1:10000
H(i,j)=1/(i+j-1);
end
end
toc
```

方法 2：

```
tic
for j=1:10000
for i=1:5
J(i,j)=1/(i+j-1);
end
end
toc
```

方法 3：

```
tic
H=zeros(5,10000);
for i=1:5
for j=1:10000
H(i,j)=1/(i+j-1);
end
end
toc
```

方法 4：

```
tic
H=zeros(5,10000);
for i=1:5
H(i,:)=1./[i:i+9999];
end
toc
```

方法 5：

```
tic
[i,j]=meshgrid(1:5,1:10000);
H=1./(i+j-1);
toc
```

可见，对于同一个问题，采用不同的方法，所需的时间是截然不同的。

2.3 数值运算

MATLAB 可以进行数学领域中的多种运算，包含大量的数学运算函数，本节中表格所列的函数仅是部分常用的函数。

2.3.1 矩阵运算

矩阵运算是线性代数中极其重要的部分，MATLAB 具有强大的矩阵运算能力，常见的运算功能有

- 矩阵的行列式；
- 矩阵的四则运算；
- 矩阵的幂和平方根；
- 矩阵的指数和对数；
- 矩阵的翻转；
- 矩阵的逆运算；
- 矩阵的迹；
- 矩阵的范数；
- 矩阵的条件数；
- 矩阵的重塑；
- 矩阵的逻辑运算；
- 矩阵的初等变换；
- 矩阵的秩。

矩阵运算的函数非常多，在此不一一列举。

【例 2-3-1】先建立 5×5 矩阵 ***A***，然后将 ***A*** 的第一行元素乘以 1，第二行乘以 2，…，第五行乘以 5。

程序如下：

```
A=[17,0,1,0,15;23,5,7,14,16;4,0,13,0,22;10,12,19,21,3;...
11,18,25,2,19];
D=diag(1:5);    %对角阵
D*A             %用 D 左乘 A,对 A 的每行乘以一个指定常数
```

【例 2-3-2】利用随机函数产生一个三阶方阵 ***A***，然后计算方阵之行列式的值。

程序如下：

```
A=rand(3)
A=
    0.9501    0.4860    0.4565
    0.2311    0.8913    0.0185
    0.6068    0.7621    0.8214
det(A)
ans=
    0.4289
```

【例 2-3-3】求矩阵 A 的三种范数。

程序如下：

```
A=[17,0,1,0,15;23,5,7,14,16;4,0,13,0,22;10,12,19,21,3;11,18,25,2,19];
a1=norm(A,1)                %求 A 的 1—范数
a2=norm(A)                  %求 A 的 2—范数
ainf=norm(A,inf)            %求 A 的∞—范数
```

【例 2-3-4】用求特征值的方法解方程 $7x^3+5x^2+2x-18=0$

程序如下：

```
p=[7,5,2,-18];
A=compan(p);                %A 的伴随矩阵
x1=eig(A)                   %求 A 的特征值
x2=roots(p)                 %直接求多项式 p 的零点
```

若一个矩阵只有少数的元素非零，则称其为稀疏矩阵。在 MATLAB 中，可以用满矩阵存储方式和稀疏矩阵存储方式来存储矩阵。

【例 2-3-5】将完全矩阵 A=[0 0 3;1 0 0;0 5 0]转化成稀疏矩阵。

程序如下：

```
X=[0 0 3;1 0 0;0 5 0]
S=sparse(X)
X=full(S)
```

2.3.2 统计分析

对工程及科学实验所测量的数据进行分析统计，是实验评估的一项重要工作。统计分析包括的内容很多，如求极值、平均值和中值、累加和与累乘积、排序、标准方差、概率分布等。

MATLAB 以数组为操作对象，因此很容易对数据集合进行统计分析。按照规定，数据集合存储在面向列的矩阵里，即矩阵的每一列代表不同的被测变量，每一行代表各个样本或观察值。常用的统计分析函数如表 2-3-1 所示。

表 2-3-1　统计分析函数

函　数　名	功 能 描 述	函　数　名	功 能 描 述
cumprod	向量累积	prod	对向量中各元素求积
cumsum	向量累加	sort	对向量中各元素排序
max	求向量中最大元素	sortrows	对矩阵中各行排序
min	求向量中最小元素	std	求向量中各元素标准差
mean	求向量中各元素均值	sum	对向量中各元素求和
median	求向量中中间元素	trapz	梯形法求数值积分

【例 2-3-6】生成满足正态分布的 10 000×5 随机矩阵，然后求各列元素的均值和标准方差，再求这 5 列随机数据的相关系数矩阵。

程序如下：

```
X=randn(10000,5)
M=mean(X)
D=std(X)
R=corrcoef(X)
```

【例 2-3-7】对矩阵进行各种排序。

程序如下：

```
A=[1,-8,5;4,12,6;13,7,-13];
sort(A)                    %对 A 的每列按升序排序
sort(-A,2)                 %对 A 的每行按降序排序
[X,I]=sort(A)              %对 A 按列排序，并将每个元素所在行号送矩阵 I
```

【例 2-3-8】求矩阵 A 的每行元素的乘积和全部元素的乘积。

程序如下：

```
A=[1,2,3,4;5,6,7,8;9,10,11,12];
S=prod(A,2)
prod(S)                    %求 A 的全部元素的乘积
```

2.3.3 多项式运算

在工程及科学分析上，多项式常被用来模拟一个物理现象的解析函数，在 MATLAB 里，多项式由一个行向量表示，多项式的系数按降幂排列。常用的多项式运算函数如表 2-3-2 所示。

表 2-3-2 多项式运算函数

函数名称	作　用	函数名称	作　用
roots(C)	求多项式的根	poly(v,s)	计算多项式的值
poly(a)	求特征多项式	polyfit(x,y,n)	n 次多项式拟合
polyder(p)	求多项式的导数	polyvalm	求矩阵多项式值
deconv	因式分解与多项式乘法	polyval	多项式求值
polyeig	多项式特征值	roots	求多项式的根

由于多项式的加法和减法与向量的加法和减法运算相似，没有提供专门的加法和减法命令。对于两个阶数不同的多项式相加，只相同阶数的相加，没有的阶数用零填补。多项式相乘是一个卷积的过程，函数为 conv，此函数允许嵌套使用；多项式相除是其逆过程，函数为 deconv。

【例 2-3-9】多项式的乘积和相除。

程序如下：

```
a=[1,3,3,1];
b=[1,1];
c=conv(a,b)
a=[1 4 6 4 3];
```

```
b=[1,2,1];
[c,d]=deconv(a,b)
```

根据多项式的系数行向量，可对多项式进行加、减、乘、除等运算，同样也能对它们进行求值，可以用函数 polyval 实现这一功能，例如 polyval(p,x)，可计算出多项式 p 在 x 的每一点值，其中 p 代表多项式各阶系数的数组，x 为数组。

可使用函数 r=roots(p)求多项式的根。若已知多项式的根时，也可以构造相应的多项式，在 MATLAB 中，poly 函数能够实现这一功能，其用法为 p=poly(r)，其中 r 是代表根的数。多项式求导数的函数是 polyder。

【例 2-3-10】求多项式 $p(x)=3x^2+2x+1$ 在点 $x=5$，7，9 处的值。

程序如下：

```
p=[3 2 1]
polyval(p,[5 7 9])
```

【例 2-3-11】多项式求导数。

```
a=[2,1];
b=[3,6,7];
k1=polyder(b)
k2=polyder(a,b)
[q,d]=polyder(b,a)
```

【例 2-3-12】用一个 5 次多项式在区间[0，2π]内逼近函数 sin(x)。

程序如下：

```
X=linspace(0,2*pi,50);Y=sin(X);
[P,S]=polyfit(X,Y,5)                          %得到 5 次多项式的系数和误差
plot(X,Y,'k*',X,polyval(P,X),'k-')
```

2.3.4 解方程

方程的形式众多，其求解的方法也很多。例如，针对线性方程，其求解可分为两类：

（1）方程组求唯一解或求特解，可以用直接法解低阶稠密矩阵，用迭代法解大型稀疏矩阵。

（2）方程组求无穷解，即通解，可以通过系数矩阵的秩来判断，若系数矩阵的秩 $r=n$（n 为方程组中未知变量的个数），则有唯一解；若系数矩阵的秩 $r<n$，则可能有无穷解。

【例 2-3-13】求解方程组：

$$\begin{cases} x^2+y-6=0 \\ y^2+x-6=0 \end{cases}$$

程序如下：

```
 clear;
[x,y]=solve('x^2+y-6=0', 'y^2+x-6=0','x','y')
```

若将[x,y]=改用 X= ，则仅将返回一个解的结构。

【例 2-3-14】求解方程组：

$$\begin{cases} 5x_1+4x_3+2x_4=3 \\ x_1-x_2+2x_3+x_4=1 \\ 4x_1+x_2+2x_3=1 \\ x_1+x_2+x_3+x_4=0 \end{cases}$$

程序如下：

```
clear;
A=[5,0,4,2;1,-1,2,1;4,1,2,0;1,1,1,1];
b=[3;1;1;0];    X=linsolve(A,b)
```

2.3.5 曲线拟合与插值

在大量应用中，常面临用一个解析函数描述数据间的关系问题。解决这个问题有两种方法，即插值和曲线拟合，常用的函数如表 2-3-3 所示。插值是确定某个函数在两个采样值之间的数值时采用的运算过程，通常利用曲线拟合的方法，即通过离散的输入采样点建立一个连续函数，用这个重建的函数便可以求出任意位置处的函数值。

表 2-3-3 曲线拟合与插值函数

函 数 名	功 能 描 述	函 数 名	功 能 描 述
griddata	数据网络的插值生成	interpft	一维插值（FFT 方法）
interp1	一维插值（查表）	interpn	多维插值（查表）
interp2	二维插值（查表）	meshgrid	构造三维图形用 x，y 阵列
interp3	三维插值（查表）	spline	三次样条插值
polyfit	求最小二乘拟合多项式系数	—	—

要找出近似一组数据（也就是最能代表这些数据）的曲线方程组，有许多选择，例如从简单的一阶线性方程组到高阶的多项式。

MATLAB 的 polyfit 函数提供了从一阶到高阶多项式的回归法，用 polyfit 函数来求得最小二乘拟合多项式的系数，再用 polyval 函数按所得的多项式计算所给出的点上的函数近似值。polyfit 函数的调用格式为

```
[P,S]=polyfit (X,Y,m)
```

函数根据采样点 X 和采样点函数值 Y，产生一个 m 次多项式 P 及其在采样点的误差向量 S。其中 X,Y 是两个等长的向量，P 是一个长度为 m+1 的向量，P 的元素为多项式系数。polyval 函数的功能是按多项式的系数计算 x 点多项式的值。

【例 2-3-15】在一天 24 小时内，从零点开始每间隔 2 小时测得的环境数据温度分别为“12,9,9,10,18,24,28,27,25,20,18,15,13”，推测中午一点时的温度。

程序如下：

```
clear;x=0:2:24;
y=[12,9,9,10,18,24,28,27,25,20,18,15,13];
y1=interp1(x,y,13,'spline')
```

【例 2-3-16】考虑如下 x 和 y 一组实验数据，分别用多种方式进行曲线拟合。

```
x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y=[1.2, 3, 4, 4, 5, 4.7, 5, 5.2, 6, 7.2]
```

一次多项式拟合：

```
p1 = polyfit(x,y,1)
```

三次多项式拟合：

```
p3 = polyfit(x,y,3)
```

plot 原始数据、一次拟合曲线和三次拟合曲线：

```
x2=1:0.1:10;
y1=polyval(p1,x2)
y3=polyval(p3,x2)
plot( x, y, '*', x2, y1, ':', x2, y3)
```

由图 2-3-1 可见，三次拟合结果较好。

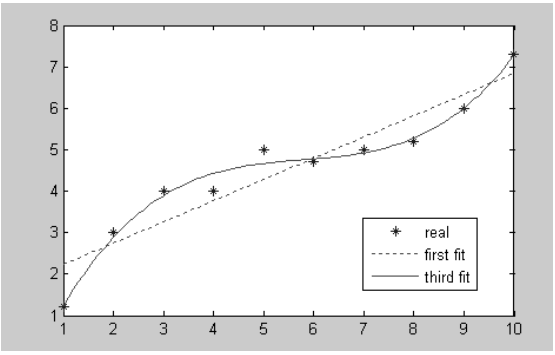


图 2-3-1　例 2-3-16 图

2.3.6　微积分

包括求极限、求导、微分、积分运算等。

【例 2-3-17】设 x 由[0，2π]间均匀分布的 10 个点组成，求 sin$x$ 的 1～3 阶差分。程序如下：

```
X = linspace(0,2*pi,10);
Y = sin(X);
DY = diff(Y)
D2Y = diff(Y,2)
D3Y = diff(Y,3)
```

【例 2-3-18】求定积分。

$$k=\int_0^{\pi}\sin(x)\mathrm{d}x=-\cos x\Big|_0^{\pi}=2$$

程序如下：

```
x=0:pi/100:pi;
y=sin(x);
k=trapz(x,y)
```

2.3.7 概率统计

包括生成随机数、概率分布及方差分析等。

【例 2-3-19】设随机变量 X 的分布密度为

$$f(x)=\begin{cases}\dfrac{2}{\pi}\cos 2x, & |x|\leqslant\dfrac{\pi}{2}\\ 0, & \text{其他}\end{cases}$$

求随机变量 X 的期望和方差。

程序如下：

```
clear;syms  x;fx=2/pi*(cos(x))^2;
EX=int(x*fx,x,-pi/2,pi/2)
E2X=int(x^2*fx,x,-pi/2,pi/2)
DX=E2X-EX^2
```

【例 2-3-20】设生成一组均值为 15，方差为 2.52 的正态分布的随机数据，然后对这组数据进行置信度 97%的参数估计。

程序如下：

```
clear;
w=normrnd(15,2.5,50,1);    或 w=15+2.5*randn(50,1);
alpha=0.03;
[mh,sh,mc,sc]=normfit(w,alpha)
```

【例 2-3-21】设从一大批产品中抽取 100 个产品，经检验知有 60 个一级品，求这批产品的一级品率（置信度 95%）。

```
clear; alpha=0.05;N=100;X=60;
[Ph,Pc]=mle('bino',X,alpha,N)
```

2.4 符号运算

符号运算是指在运算时无须事先对变量赋值，而将所得到结果以标准的符号形式来表示。 MATLAB 符号运算以 Maple 的内核作为符号计算引擎，依赖 Maple 已有的函数库，开发了符号计算工具箱，可以完成几乎所有的符号运算功能，包括符号表达式的运算，符号表达式的复合、化简，符号矩阵的运算，符号微积分、符号作图，符号代数方程求解，符号微分方程求解等。

MATLAB 符号运算有以下特点：

- 计算以推理方式进行，因此不受计算误差累积所带来的困扰；
- 符号计算中出现的数字也都是当成符号处理；
- 符号计算所需的运行时间相对较长。

2.4.1 符号变量和表达式

MATLAB 提供了两个建立符号对象的函数：sym 和 syms，sym 函数用来建立单个符号量，syms 一次可以定义多个符号变量。

含有符号对象的表达式称为符号表达式，建立符号表达式有以下三种方法：

（1）利用单引号来生成符号表达式。

```
f='sin(x)+1'
```

（2）用 sym 函数建立符号表达式。

```
y=sym('sin(x)+cos(x)')
```

（3）使用已经定义的符号变量组成符号表达式。

```
x=sym('x');
y=sin(x)+cos(x)
```

符号表达式中自由变量的确定规则为：

- 只对除 i 和 j 以外的单个小写英文字母进行搜索；
- 默认 x 是首选变量；
- 其余小写字母被选中为自由变量的次序是，在字母表中，靠 x 距离近的优先，在 x 之后的优先。

函数 sym 可以将数值表达式变换成符号表达式，函数 numeric 或 eval 可以将符号表达式变换成数值表达式。

2.4.2 符号运算实例

在 MATLAB 中，大多数值运算的功能和函数也可直接运用于符号运算，只需在运算前需定义符号变量和表达式。也有一些专用于符号运算的函数，常用的符号运算函数如表 2-4-1 所示。

表 2-4-1　符号运算函数

函　数	功　能
sym(‘x’)	创建一个符号变量 x
syms a b c …	创建多个符号变量 x
findsym	对默认自变量进行查询
factor(S)	因式分解
expand(S)	因式展开
collect(S,n)	将符号表达式 S 中自变量 n 的同次幂项的系数合并
[r,how]=simple(S)	返回 S 的最简化形式

（续表）

函　数	功　能
[n,d]=numden(S)	将符号表达式 S 转换为分子和分母都是整系数的最佳多项式
horner(S)	将符号表达式 S 转换为嵌套形式
limit(F)	计算符号表达式 F 在默认自变量趋向于 0 时的极限
diff(S,n)	求符号表达式 S 对于自变量 v 的微分
int(S,a,b)	求符号表达式 S 对于默认自变量从 a 到 b 的定积分
symsum(S)	求符号表达式 S 对于默认自变量的不定和
taylor(f)	计算符号表达式 f 在自变量 v=0 处的 n–1 阶 Taylor 级数展开式
solve(eq)	求解符号表达式 eq=0 的代数方程，自变量为默认自变量
dsolve('eq1,eq2,…'.'cond1,cond2,…','v')	求由 eq1,eq2,…指定的常微分方程的符号解，参数 cond1,cond2,…为指定常微分方程的边界条件和初始条件，自变量 v 如果不指定，将为默认自变量

1．因式分解与展开

MATLAB 提供了符号表达式的因式分解与展开的函数，函数的调用格式有

- factor(s)：对符号表达式 s 分解因式；
- expand(s)：对符号表达式 s 进行展开；
- collect(s)：对符号表达式 s 合并同类项；
- collect(s,v)：对符号表达式 s 按变量 v 合并同类项。

【例 2-4-1】因式分解。

程序如下：

```
syms x y;
s=(-7*x^2-8*y^2)*(-x^2+3*y^2);
expand(s)                          %对 s 展开
collect(s,x)                       %对 s 按变量 x 合并同类项(无同类项)
factor(ans)                        %对 ans 分解因式
```

2．化简

MATLAB 提供的对符号表达式化简的函数有

- simplify(s)：应用函数规则对 s 进行化简；
- simple(s)：调用 MATLAB 的其他函数对表达式进行综合化简，并显示化简过程。

例如：

```
syms x y;s=(x^2+y^2)^2+(x^2-y^2)^2;
simple(s)                  %MATLAB 自动调用多种函数对 s 进行化简,并显示每步结果
```

3．符号矩阵

符号矩阵也是一种符号表达式，所以前面介绍的符号表达式运算都可以在矩阵意义下进行。但应注意这些函数作用于符号矩阵时，是分别作用于矩阵的每一个元素的。

由于符号矩阵是一个矩阵，所以符号矩阵还能进行有关矩阵的运算。MATLAB 还有一些专用于符号矩阵的函数，这些函数作用于单个的数据无意义。例如

- transpose(s)：返回 s 矩阵的转置矩阵；

- determ(s)：返回 s 矩阵的行列式值。

许多应用于数值矩阵的函数，如 diag、triu、tril、inv、det、rank、eig 等，也可直接应用于符号矩阵。

4．函数复合

在数学领域，两个函数的复合函数是指一个将第一个函数作用于参数，然后将第二个函数作用于结果的函数。

【例 2-4-2】将 $f=x^t$ 和 $x=\tan(y)$ 复合到一个函数中，指定 x 和 y 为它们的独立变量，自变量为 z。

程序如下：

```
syms x y t z;
g=tan(y);
f=x^t;
compose(f,g,x,y,z)
```

5．解方程

1）符号代数方程

在 MATLAB 中，求解符号代数方程可由函数 solve 实现

2）符号常微分方程

符号常微分方程求解可以通过函数 dsolve 来实现，方程中用大写字母 D 表示导数。例如，Dy 表示 y'，D2y 表示 y''，Dy(0)=5 表示 $y'(0)=5$。D3y+D2y+Dy–x+5=0 表示微分方程 $y'''+y''+y'-x+5=0$。调用格式为：

```
dsolve (e,c,v)
```

该函数求解常微分方程 e 在初值条件 c 下的特解。参数 v 描述方程中的自变量，省略时按缺省原则处理，若没有给出初值条件 c，则求方程的通解。

【例 2-4-3】解方程 $f=ax^2+bx+c$。

程序如下：

```
f='a*x^2+b*x+c';
solve(f)                          %对缺省变量 x 求解
ans=
[1/2/a*(-b+(b^2-4*a*c)^(1/2))]
[1/2/a*(-b-(b^2-4*a*c)^(1/2))]
```

【例 2-4-4】用微分方程的数值解法和符号解法解方程，并对结果进行比较。

程序如下：

```
y=dsolve('Dy+2*y/x-4*x','y(1)=2','x')      %用符号方法得到方程解析解
```

① 为了求方程的数值解，需要按要求建立一个函数文件 fxyy.m。

```
function f=fxyy(x,y)
```

```
f=(4*x^2-2*y)/x;                %只能是 y'=f(x,y)的形式,若不是这种形式时,要变形
return
```

② 输入命令，得到区间[1，2]中的数值解，以向量 t、w 存储。

```
[t,w]=ode45('fxyy',[1,2],2);
```

③ 为了对两种结果进行比较，在同一个坐标系中作出两种结果的图形。输入命令：

```
x=linspace(1,2,100);
y=x.^2+1./x.^2;                 %为作图把符号解的结果离散化
plot(x,y,'b.',t,w,'r-');
```

运行后的图形如图 2-4-1 所示，一阶微分方程的数值解（点）和符号解（圈）与解析解（线）符合得很好。

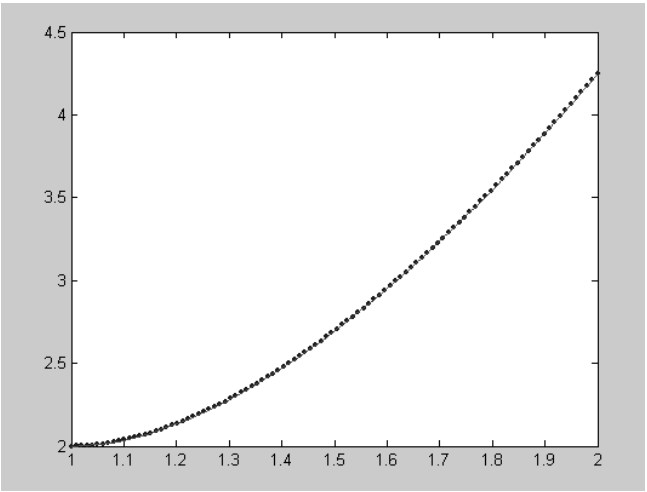


图 2-4-1　数值解与符号解的图形

6．符号微积分

在 MATLAB 中，微分和求导函数可同时处理数值和符号两种情况。

【例 2-4-5】求 $\lim_{x\to\infty}(\sqrt{x^2+x+1}-\sqrt{x^2-x+1})$。

程序如下：

```
syms x
f=sqrt(x^2+x+1)-sqrt(x^2-x+1);
a=limit(f,x,inf, 'left')
b=limit(f,x,-inf, 'right')
```

【例 2-4-6】求 $f(x)=\dfrac{(x-1)^5}{x+1}$ 的二阶导数。

程序如下：

```
syms x
f=(x-1)^5/(x+1);
```

```
df=diff(f,1);                    %求导数
d2f=diff(f,2);
df=simplify(df)%化简
d2f=simplify(d2f)
```

【例 2-4-7】求隐函数 $F(x,y)=x-y+\frac{1}{2}\sin y$ 所确定的导数 $\frac{\mathrm{d}y}{\mathrm{d}x}$。

程序如下：

```
f=sym('x-y+1/2*sin(y)');
fx=diff(f,'x');
fy=diff(f,'y');
dv=-fx/fy;
simplify(dv)%化简
```

极限可用于计算函数曲线的渐近线，导数可用于求函数的极值和拐点，判断函数的单调性等。下面结合一个具体的例子进行介绍。

【例 2-4-8】作函数 $f(x)=\frac{3x^2+6x-1}{x^2+x-3}$ 的图形，绘制出渐近线和极值（见图 2-4-2）。

程序如下：

```
syms x
num=3*x^2+6*x-1;
denom=x^2+x-3;
f=num/denom;
a=limit(f,inf);
b=double(a);
roots=solve(denom);
ezplot(f)                                    %符号函数作图命令。
hold on                                      %在原有的图形上面叠加图形。
plot([-2*pi 2*pi],[b b], 'g')                %绘水平渐近线
plot(double(roots(1))*[1 1],[-5 10], 'r') %绘垂直渐近线
plot(double(roots(2))*[1 1],[-5 10], 'r') %绘垂直渐近线
hold off                                     %取消图形叠加
f1=diff(f);                                  %求一阶导数
f1=simplify(f1);                             %化简
crit_pts=solve(f1);                          %求驻点
hold on
plot(double(crit_pts),double(subs(f,crit_pts)),'ro');
title('渐近线和极值')                        %加标题
text(-5,3,'极小值')                          %加标注
text(-2,2,'极大值')
hold off
```

【例 2-4-9】求不定积分 $\int x^n\mathrm{d}x$。

程序如下：

```
syms x n
f=x^n;
F=int(f,x)%求不定积分
```

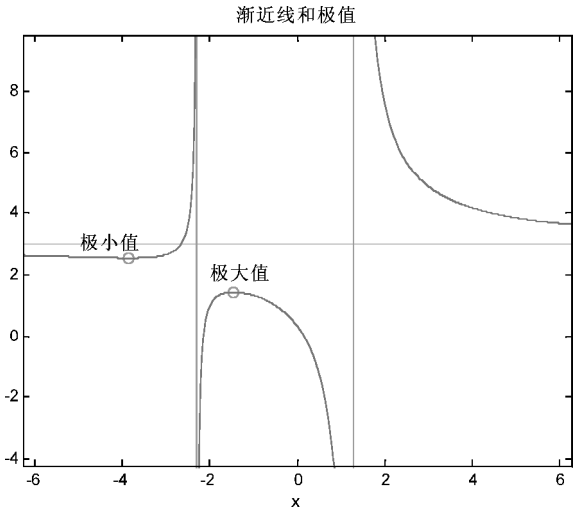


图 2-4-2　渐近线和极值

【例 2-4-10】求定积分 $\int_0^1 x^9 \mathrm{d}x$。

程序如下：

```
syms x n
f=x^9;
A=int(f,x,0,1)                    %求定积分
```

在实际应用中，往往需要对定积分进行近似计算。常用的近似计算方法有矩形法、梯形法和抛物线法等。用 MATLAB 提供的 trapz 函数可以用梯形法近似求取定积分的值。

【例 2-4-11】求定积分 $\int_0^2 \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2}} \mathrm{d}x$ 的近似值。

程序如下：

```
b=2;
x=linspace(0,b,10000);
y=exp(-x.^2./2)./sqrt(2*pi);
A=trapz(x,y)                      %求定积分的近似值
```

【例 2-4-12】计算由两条抛物线 $y^2=x$ 和 $y=x^2$ 所围成的图形的面积。

程序如下：

```
%求曲线的交点
[x1,y1]=solve('y^2=x','y=x^2');
x1=double(x1);y1=double(y1);
n=numel(x1);
%下面寻找实数解
```

```
m=1;x0=[];y0=[];
for k=1:n
    if isreal(x1(k))&&isreal(x1(k))
        x0(m)=x1(k);y0(m)=y1(k);
        m=m+1;
    end
end
x0=sort(x0);y0=sort(y0);                    %排序
%下面计算定积分
syms x
f=sqrt(x)-x^2;
A=int(f,x,x0(1),x0(2))
```

【例 2-4-13】计算由曲线 $y=\frac{2}{3}x^{3/2}$ 上相应于 x 从 a 到 b 的一段弧的长度。

程序如下：

```
syms x a b
f=2*x^(3/2)/3;
d=diff(f);
g=sqrt(1+d^2);
S=int(g,x,a,b)
```

运行结果：

```
S =
2/3*(1+b)^(3/2)-2/3*(1+a)^(3/2)
```

7．级数

级数是表示函数、研究函数性质以及进行数值计算的一种重要工具，是高等数学的重要组成部分。

【例 2-4-14】求级数 $\sum_{i=0}^{n-1}k$、$\sum_{k=1}^{\infty}\frac{1}{k^2}$ 和 $\sum_{k=0}^{\infty}x^k(|x|<1)$。

程序如下：

```
syms x k n
s1=symsum(n)
s2=symsum(1/k^2,1,inf)
s3=symsum(x^k,k,0,inf)
```

泰勒级数展开函数的调用格式如下：

```
taylor(f,n,v,a)
```

返回 f 关于 a 的 n–1 阶泰勒级数近似。变量 a 可以是数值、符号或表示数值值或未知值的字符串，n、v 和 a 的顺序没有先后之分。taylor 函数根据变量的位置和类型确定它们的用途。还可以忽略 n、v、a 等变量中的任何一个，如果不确定 v，taylor 函数可用 findsym 函数确定函数的独立变量。n 的默认值为 6。

【例 2-4-15】求 $f(x)=\dfrac{1}{5+4\cos(x)}$ 的泰勒级数展开，取前 7 项。

程序如下：

```
syms x
f=1/(5+4*cos(x))
T=taylor(f,8)
```

返回

```
T=
                1/9+2/81*x^2+5/1458*x^4+49/131220*x^6
```

8．数学变换及分析

常见的积分变换有傅里叶变换、拉普拉斯变换和 Z 变换。

（1）傅里叶（Fourier）变换。

```
fourier (f,x,t)
```

求函数 f(x)的傅里叶像函数 F(t)。

```
ifourier (F,t,x)
```

求傅里叶像函数 F(t)的原函数 f(x)。

（2）拉普拉斯（Laplace）变换。

```
laplace (fx,x,t)
```

求函数 f(x)的拉普拉斯像函数 F(t)。

```
ilaplace (F,t,x)
```

求拉普拉斯像函数 F(t)的原函数 f(x)。

（3）Z 变换。

```
ztrans (fn,n,z)
```

求 fn 的 Z 变换像函数 f(z)。

```
iztrans (Fz,z,n)
```

求 Fz 的 z 变换原函数 f(n)。

【例 2-4-16】使用 fourier 和 ifourier 函数对符号表达式 sin(*x*)进行积分变换。

程序如下：

```
syms x
f1=sin(x);
ff1=fourier(f1)     %fourier 变换
if1=ifourier(ff1)   %fourier 反变换
```

第 3 章　数据可视化及 GUI

可视化技术一直是数学计算人员所喜爱的和追求的一项技术，因为不管是数值计算还是符号计算，一般很难直接从大量的数据堆或符号堆中感受它们的具体含义。人们更喜欢直接用眼睛看到直观的图形。MATLAB 包含有丰富的图形处理能力，包括高级的二维、三维数据可视化、图像处理、模拟、图形表示等图形命令，还包括低级的图形命令，供用户自由制作、控制图形特性之用。

3.1　数据可视化

MATLAB 语言不仅具有高层绘图能力，而且还具有底层绘图能力，具有从简单的点、线、面处理发展到集二维图形、三维图形甚至四维表现图和对图进行着色、消隐、光照、渲染及多视角处理等多项功能。常用的绘图函数如表 3-1-1 所示。

表 3-1-1　常用的绘图函数

函　数	功　能	函　数	功　能
plot	绘制二维曲线图	plot3	绘制三维曲线图
mesh	绘制三维网格图	surf	绘制三维曲面图
meshc	将网格与等高线结合	surfc	带等高线的曲面图
meshgrid	生成网格点	surfl	单元颜色平滑处理的曲面
meshz	屏蔽的网格图	fplot	自适应曲线图
ezplot	二维符号函数图	ezplot3	三维符号函数图
ezmesh	函数网格图	ezsurf	函数曲面图

3.1.1　二维及三维图形绘制

1. 二维绘图

二维图形的绘制是 MATLAB 语言图形处理的基础。基本绘图函数 plot 有多种格式，能绘制单条或多条曲线，设置线型及颜色。

【例 3-1-1】绘制二维曲线，观察 plot 函数的用法。

程序如下：

```
subplot(1,3,1);
x=linspace(0,2*pi,30); %生成一组线性等距数值
y=sin(x);
plot(x,y)              %生成 30 个点连成的光滑的正弦曲线
subplot(1,3,2);
```

```
y=[0 0.55 0.70 0.90 0.83 0.25];
plot(y)                    %以序号为横坐标、数组 y 的数值为纵坐标绘制折线
subplot(1,3,3);
x=[1,2,3;4,5,6;7,8,9];y=[2,4,5;3,6,7;4,6,8]
plot(x,y)   %以 x 的第 i 列分量为横坐标， y 的第 i 列分量为纵坐标，绘制 i 条连线
```

例 3-1-1 的结果如图 3-1-1 所示。

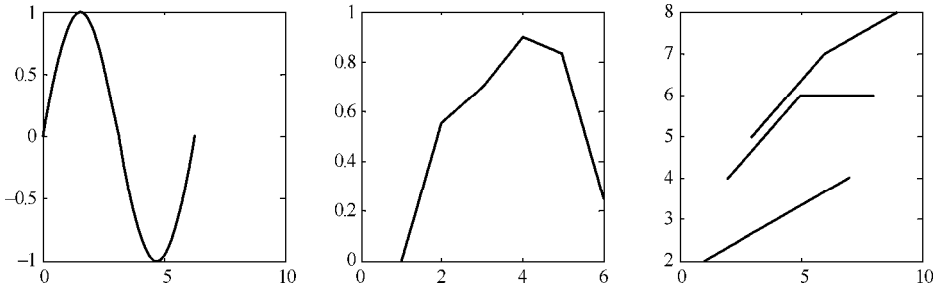


图 3-1-1　二维曲线函数

【例 3-1-2】绘制衰减振荡曲线函数 $y=e^{-0.5}\sin 5x$ 图形。

程序如下：

```
x=0:0.1:4*pi; y= exp(-0.5*x) ;y1=y .*sin(5*x);
plot(x,y1,x,y, '-r',x,-y,'-r')     %在同一窗口绘制多条曲线，并设定线形，颜色
```

例 3-1-2 的结果如图 3-1-2 所示。

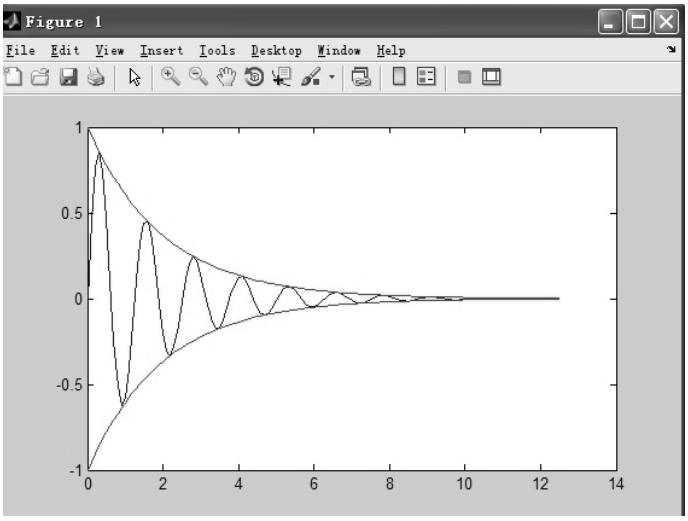


图 3-1-2　衰减振荡曲线函数

plot 命令是将从外部输入或通过函数数值计算得到的数据矩阵转化为连线图。在实际应用中，可能会因为自变量的取值间隔不合理而使曲线图形不能反映出自变量在某些区域内函数值的变化情况。MATLAB 提供了一个绘图函数 fplot，通过内部的自适应算法来动态决定自变量的取值间隔，并且确保在输出的图形中表示出所有的奇异点。

格式：

```
fplot(fun,lims,'corline')
```

fun 是 M 文件函数的名称或句柄，或者含有变量 x 的字符串。

例如：

```
f=inline('x.*sin(1./x)');
fplot(f,[-.5,.5],400);
```

2．三维绘图

二维图形的所有基本特性在三维中仍都存在。三维图形常见的表达方式有三维曲线图、三维网格图和三维曲面图。

三维曲线图的绘图函数 plot3，除了包括第三维的信息（Z 方向）之外，用法与 plot 类似。

MATLAB 在绘制函数 z=f(x,y)的网格图时，首先将其定义域 D 分为若干个小矩形（或三角形），然后计算出网格点上的函数值；最后连接相邻的函数值空间数据并构成函数的网格曲面。必须先用函数 meshgrid 生成平面网格点，然后调用 mesh 绘制图形。

三维曲面图利用函数 surf 不同颜色对网格图中的单元进行填充。

【例 3-1-3】分别绘制蓝宝石项链图和螺旋线图。

程序如下：

```
subplot(1,2,1);
t=(0:0.02:2)*pi;
x=sin(t);y=cos(t);
z=cos(2*t);
plot3(x,y,z,'b-',x,y,z,'bd')
subplot(1,2,2);
t=0:0.1:8*pi;
plot3(sin(t),cos(t),t)
title('绘制螺旋线')
```

例 3-1-3 的结果如图 3-1-3 所示。

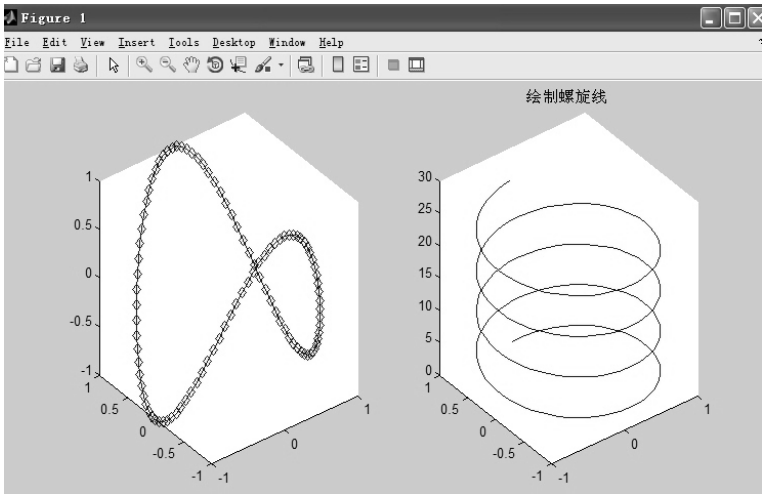


图 3-1-3　蓝宝石项链图和螺旋线图

【例 3-1-4】绘制函数 $z=x^2+y^2$ 的曲面。

程序如下：

```
x=-4:4;y=x;
[x,y]=meshgrid(x,y);              %生成 x-y 坐标"格点"矩阵
z=x.^2+y.^2;                      %计算格点上的函数值
subplot(1,2,1), mesh(x,y,z);      %三维网格图
subplot(1,2,2), surf(x,y,z);      %三维曲面图
colormap(hot);
```

例 3-1-4 的结果如图 3-1-4 所示。

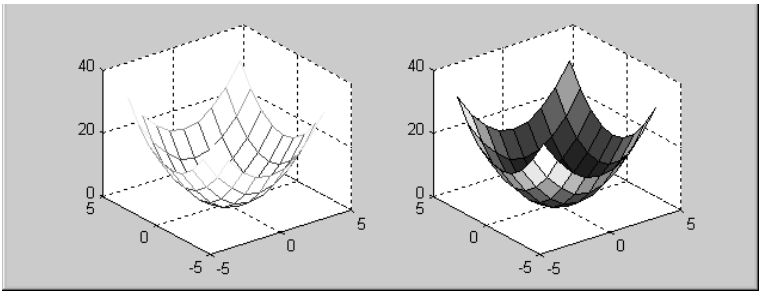


图 3-1-4　函数 $z=x^2+y^2$ 的曲面的绘制结果

【例 3-1-5】绘制山峰网格图和曲面图

程序如下：

```
[X,Y,Z]=peaks(30)                 %peaks 绘制山峰函数
subplot(2,2,1)
mesh(Z)
subplot(2,2,2)
surf(X,Y,Z)
subplot(2,2,3)
meshc(Z)
subplot(2,2,4)
surfl(X,Y,Z)
```

例 3-1-5 的结果如图 3-1-5 所示。

3．符号函数绘图

ezplot 和 ezplot3 是符号函数的曲线图，符号函数可以以隐函数表示。

例如：

```
ezplot3('exp(t/10)','sin(t)*cos(t)','t',[0,6*pi])
```

ezmesh 绘制符号函数的网格图，ezsurf 绘制符号函数表示的曲面图。

【例 3-1-6】绘制符号函数。

程序如下：

```
ezmesh('y^2-3*x*y+x^2',[-4,4,-4,4])
```

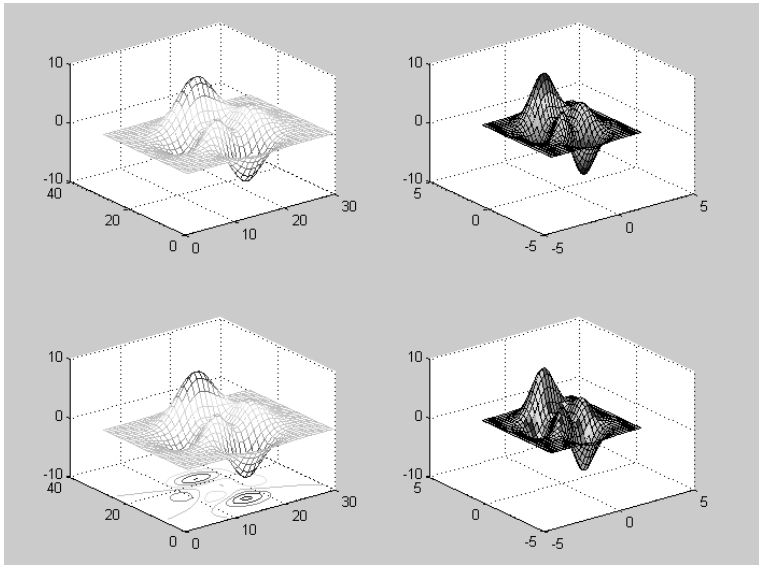


图 3-1-5　山峰网格图和曲面图

例 3-1-6 的结果如图 3-1-6 所示。

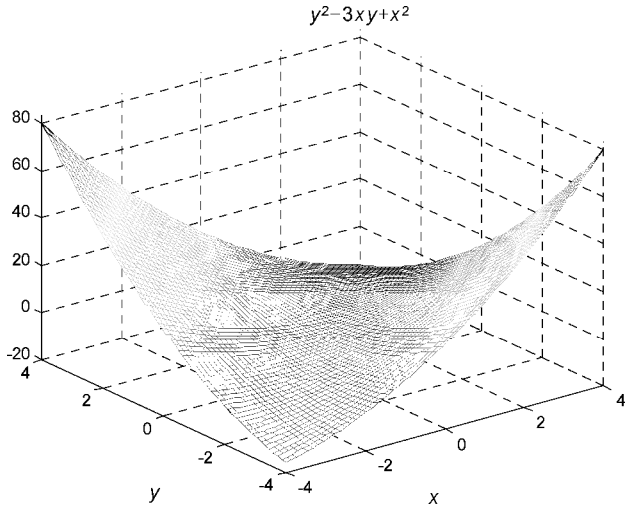


图 3-1-6　符号函数绘图

3.1.2　图形窗口的控制及修饰

1．图形窗口的控制

图形窗口的处理有多种方式。

1）图形加注功能

图形加注是指将标题、坐标轴标记、网格线及文字注释加注到图形上，常用的函数有

- title：给图形加标题；

- xlable：给 *x* 轴加标注；
- ylable：给 *y* 轴加标注；
- text：在图形指定位置加标注；
- gtext：将标注加到图形任意位置；
- grid on(off)：打开、关闭坐标网格线；
- legend：添加图例；
- axis：控制坐标轴的刻度。

【例 3-1-7】绘制正弦和余弦曲线，并加适当标注。

程序如下：

```
t=0:0.1:10
y1=sin(t);y2=cos(t);plot(t,y1,'r',t,y2,'b--');
x=[1.7*pi;1.6*pi];
y=[-0.3;0.8];
s=['sin(t)';'cos(t)'];
text(x,y,s);
title('正弦和余弦曲线');
legend('正弦','余弦')
xlabel('时间 t'),ylabel('正弦、余弦')
grid
axis square
```

例 3-1-7 的结果如图 3-1-7 所示。

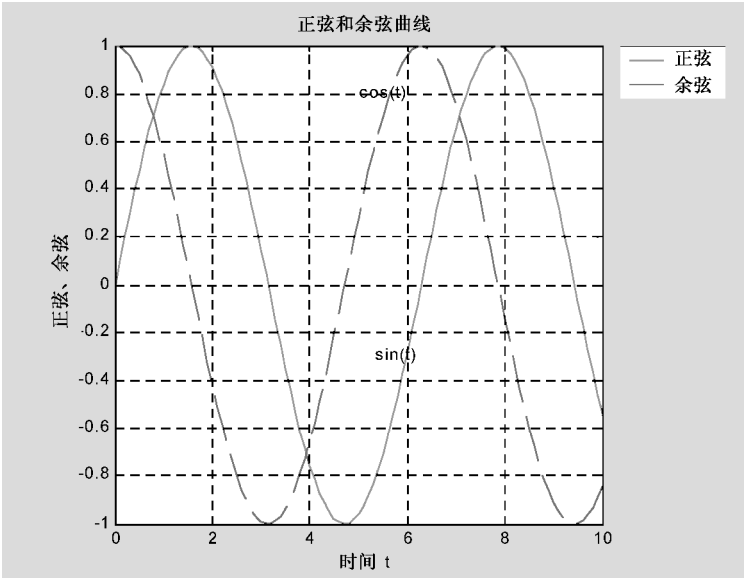


图 3-1-7　正弦和余弦曲线

2）图形保持

hold on/off：图形的保持/释放。

3）新建窗口及分割窗口

- figure：新建窗口；
- subplot(m,n,i)：把图形窗口分割为 m 行 n 列子窗口，并选定第 i 个窗口为当前窗口。

4）缩放图形

zoom on/off：图形缩放。

2．图形修饰

图形修饰通常包括颜色修饰、透视与消隐、视角修饰、裁剪修饰、等高线修饰等方法。

1）颜色的修饰

对于绝大多数的线图函数，如 plot、plot3、contour 等，一般不需要颜色映像来控制其色彩显示，而对于面图函数，如 mesh、surf 等，则需要调用色图设定函数 colormap(MAP)。MAP 为 $m\times3$ 维色图矩阵。可根据需要任意生成，也选用 MATLAB 配备的基本色调。

colorbar 函数在当前的图形中显示颜色标尺，用来反映当前使用的颜色映像，以此反映图形中数据的相对大小。

- shading faceted：网格修饰，缺省方式；
- shading flat：去掉黑色线条，根据小方块的值确定颜色；
- shading interp：颜色整体改变，根据小方块四角的值差补过度点的值确定颜色。

【例 3-1-8】绘制山峰图并进行颜色修饰。

程序如下：

```
peaks(30);shading interp;colormap(hot)
[X,Y,Z]=peaks(30);surfl(X,Y,Z)
shading interp;colormap(cool);axis off
peaks(30);colormap(hot);colorbar('horiz')
figure(2);colormap(cool);
```

例 3-1-8 的结果如图 3-1-8 所示。

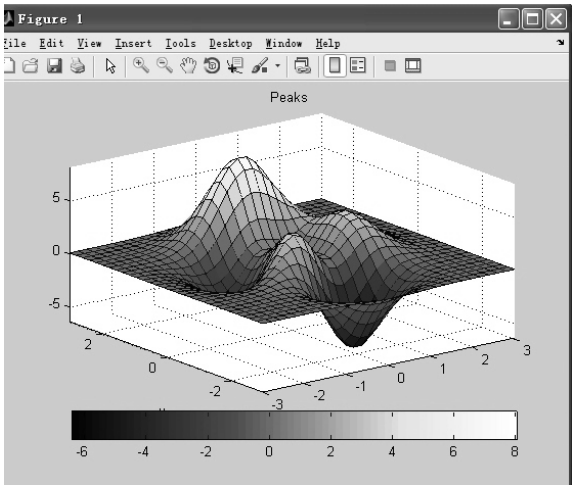


图 3-1-8　山峰网的颜色修饰

2）图形效果修饰

- 透视与消隐：用于网线图，指令为 hidden on/off；
- 裁剪修饰：用于网线图、曲面图，通过计算裁剪矩阵数值实现；
- 视角修饰：观察不同角度的三维视图，函数 view；缺省值为：方位角=–37.5，俯视角=30；
- 水线修饰：waterfall；
- 等高线修饰：contour(Z,n)用于针对矩阵 Z 绘制 n 条等高线，C=contourc(Z,n)用于计算 n 条等高线的坐标，clable(c)用于给等高线加标注。

【例 3-1-9】绘制带等位线可透视 peaks 多峰函数及裁剪图形。

程序如下：

```
subplot(1,2,1);
meshc(peaks(20))                          %绘带等位线的多峰图
colormap([1 0 0])                         %红色网线[RGB]
hidden off                                %可透视
subplot(1,2,2);
hidden on
p=peaks;
p(30:40,20:30)=nan*p(30:40,20:30);        %裁剪修饰
surf(p)
```

例 3-1-9 的结果如图 3-1-9 所示。

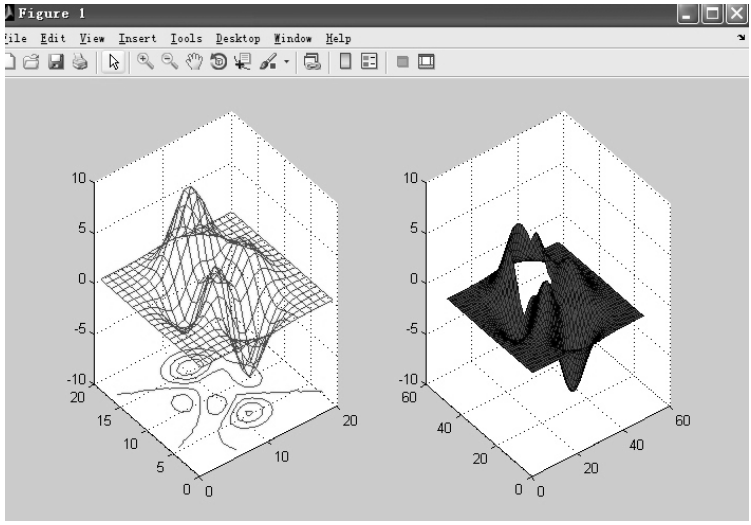


图 3-1-9　透视图及裁剪图

【例 3-1-10】观察不同视角的波峰图形。

程序如下：

```
z=peaks(40);
subplot(2,2,1);mesh(z);
subplot(2,2,2);mesh(z);view(-15,60);
```

```
subplot(2,2,3);mesh(z);view(-90,0);
subplot(2,2,4);mesh(z);view(-7,-10);
```

例 3-1-10 的结果如图 3-1-10 所示。

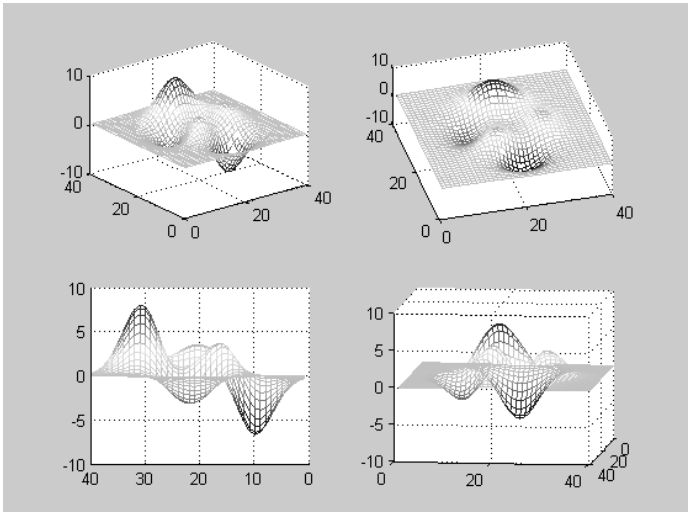


图 3-1-10　不同视角的山峰图

【例 3-1-11】分别在二维及三维坐标中绘制 peaks 函数的 10 条等高线。程序如下：

```
subplot(1,2,1);
contour(peaks,10);
subplot(1,2,2);
contour3(peaks,10)
```

例 3-1-11 的结果如图 3-1-11 所示。

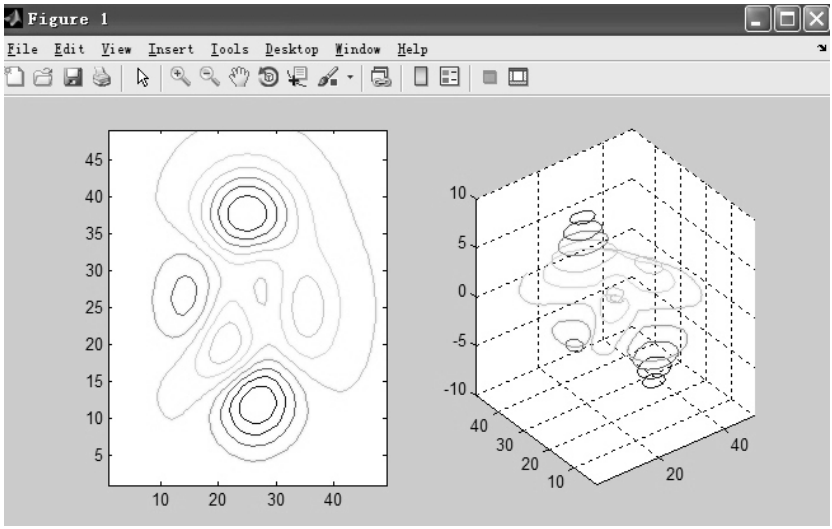


图 3-1-11　山峰网格图和曲面图

3.1.3 特殊图形绘制

特殊图形包含统计分析的直方图、饼图等，用于特殊坐标的极坐标图、对数坐标图以及常用的三维柱面图、球面图等。特殊图形绘图函数如表 3-1-2 所示。

表 3-1-2 特殊图形绘图函数

函 数	功 能	函 数	功 能
area	填充的二维图形	plotyy	双轴图
bar	条形图	polar	极坐标图
pie	饼图	pareto	帕累托图
errorbar	误差条图	stem	火柴杆图
scatter	散点图	stairs	阶梯图
hist	直方图	rose	玫瑰花图
loglog	对数坐标图	semilog	半对数坐标图
semilogx	半对数坐标图	semilogy	半对数坐标图
bar3	三维条形图	pie3	三维柄状图
comet3	三维彗星轨迹图	trisurf	三角形表面图
ezgraph3	控制绘制三维图	trimesh	三角形网格图
cplxmap	复变函数图	waterfall	瀑布图
scatter3	三维散射图	cylinder	柱面图
stem3	三维离散数据图	sphere	球面图

【例 3-1-12】求方程 $x^{10}+x^6+30x^3+1=0$ 的根并显示在极坐标上。

程序如下：

```
a=[1 0 0 0 1 0 0 30 0 0 1];
q=roots(a);
r1=abs(q);
r2=angle(q);
polar(r2,r1,'r*');
```

例 3-1-12 的结果如图 3-1-12 所示，图中所标*处为方程的根。

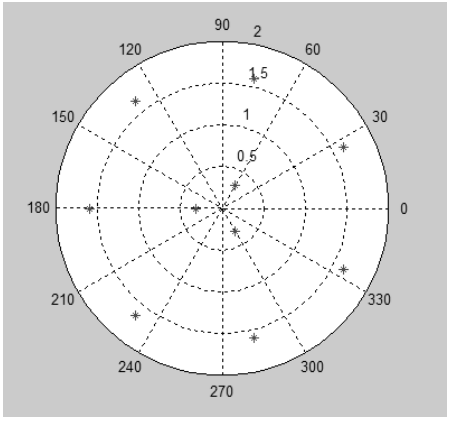


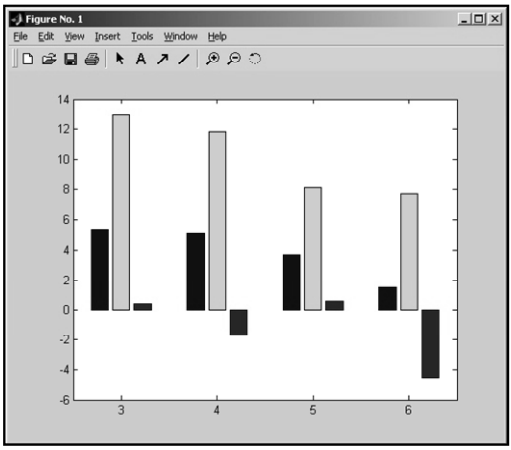
图 3-1-12 极坐标图

【例 3-1-13】用条形图表示某年一月份中 3 日～6 日连续四天的温度数据，矩阵 ***y*** 的各列分别表示平均温度、最高温度和最低温度，用条形图和三维条形图分别表示。

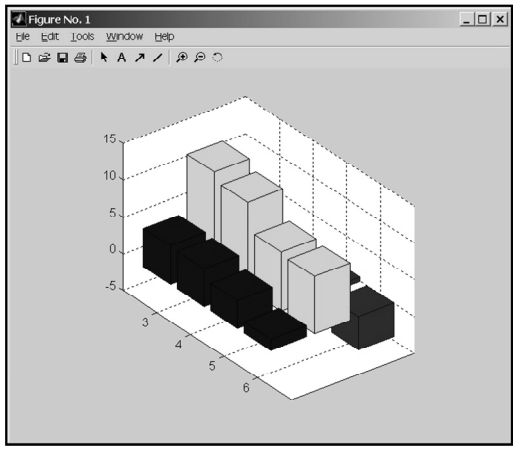
程序如下：

```
x=3:6;
y=[5.3000   13.0000    0.4000
5.1000   11.8000   -1.7000
3.7000    8.1000    0.6000
1.5000    7.7000   -4.5000]
 bar(x,y)                %画条形图
 bar3(x,y)               %画三维条形图
```

例 3-1-13 的结果如图 3-1-13 所示。



（a）条形图



（b）三维条形图

图 3-1-13　例 3-1-13 的显示效果图

由图 3-1-13 可看出条形图是按行分组的，每组为每天的平均温度、最高温度和最低温度。

【例 3-1-14】用 pie 函数生成饼图。

程序如下：

```
x=[1 3 5 7 9];
pie(x),figure
explode=[0 1 0 0 0];
pie(x,explode)          %突出显示
```

例 3-1-14 的结果如图 3-1-14 所示。

【例 3-1-15】在同一窗口中绘制阶梯图和火柴杆图，并分别填充显示。

程序如下：

```
subplot (2,2,1), stem(t,y);
title('stem(t,y)')
subplot (2,2,2), stairs(t,y);
```

```
    title('stairs(t,y)')
    subplot (2,2,3), bar(t,y);
    title('bar(t,y)')
    subplot (2,2,4), fill(t,y,'r');
    title(' fill(t,y,''r'')')
```

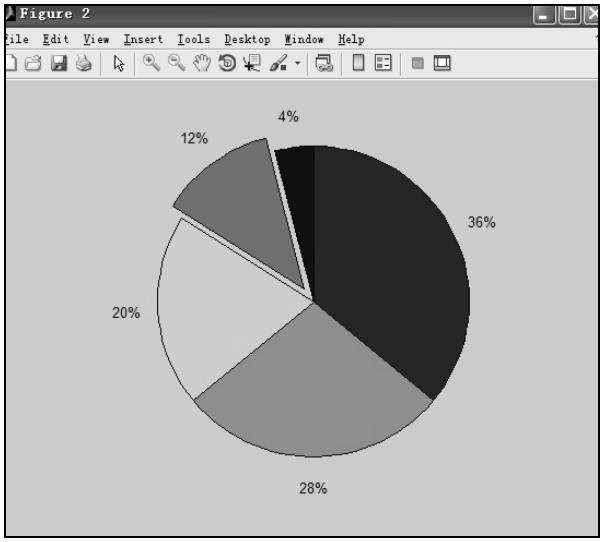


图 3-1-14　饼图

例 3-1-15 的结果如图 3-1-15 所示。

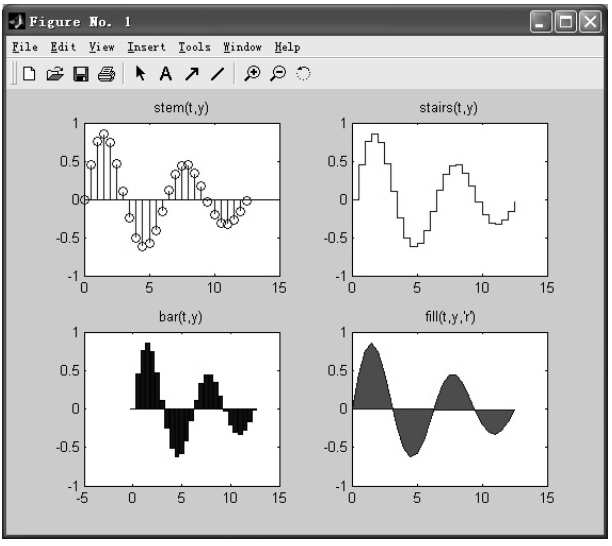


图 3-1-15　阶梯图和火柴杆图

【例 3-1-16】标准正态分布图。

程序如下：

```
    yn=randn(10000,1);  hist(yn)
    y=randn(10000,3);hist(y)
```

例 3-1-16 的结果如图 3-1-16 所示。

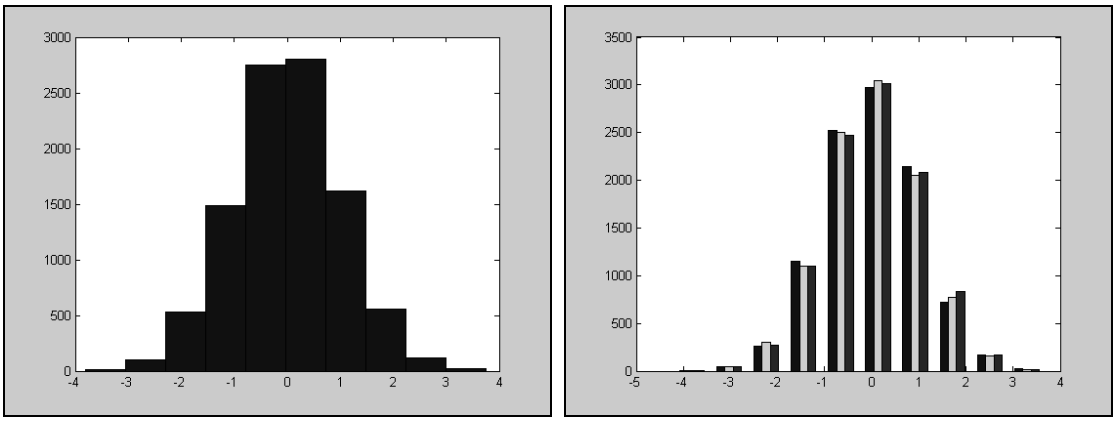


图 3-1-16　标准正态分布图

【例 3-1-17】绘制离散点图。

程序如下：

```
load seamount          %海山 scatter(x,y,50,z)
a=rand(200,1);b=rand(200,1);
c=rand(200,1);
scatter(a,b,100,c,'p')
```

例 3-1-17 的结果如图 3-1-17 所示。

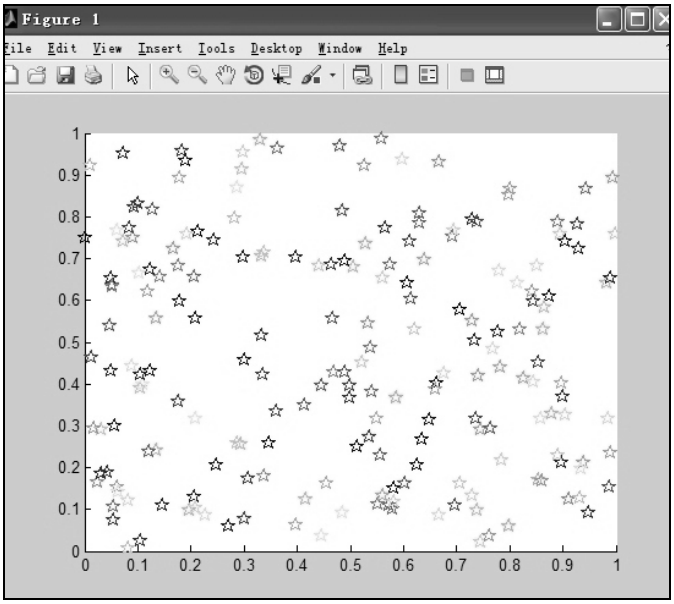


图 3-1-17　离散点图

【例 3-1-18】绘制三维陀螺锥面。

程序如下：

```
t1=0:0.1:0.9;
t2=1:0.1:2;
r=[t1 -t2+2];
[x,y,z]=cylinder(r,30);
surf(x,y,z);
grid
t=pi:0.5:3*pi;
r=sin(t)+t;
cylinder(r,50)
```

例 3-1-18 的结果如图 3-1-18 所示。

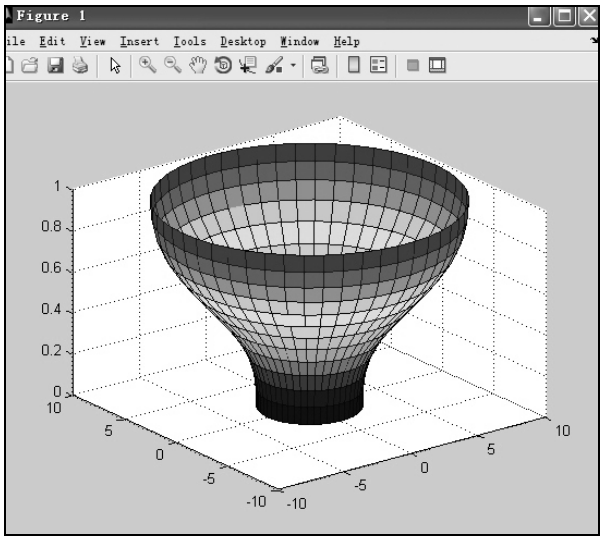


图 3-1-18　三维陀螺锥面

【例 3-1-19】不同光源光照处理后的球体。

程序如下：

```
[x,y,z]=sphere(20);
subplot(1,2,1);
surf(x,y,z);axis equal;
light('Posi',[0,1,1]);
shading interp;
hold on;
plot3(0,1,1,'p');text(0,1,1,' light');
subplot(1,2,2);
surf(x,y,z);axis equal;
light('Posi',[1,0,1]);
shading interp;
hold on;
plot3(1,0,1,'p');text(1,0,1,' light');
```

例 3-1-19 的结果如图 3-1-19 所示。

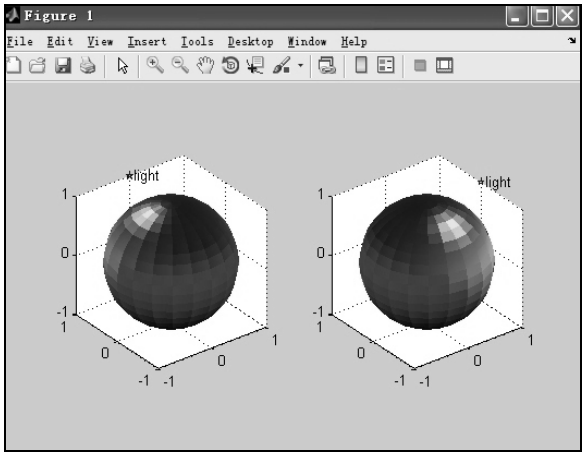


图 3-1-19　不同光源光照处理

【例 3-1-20】绘复变函数图。

程序如下：

```
z=cplxgrid(20);
cplxmap(z,z)
figure
cplxmap(z,z.^2)
figure
cplxmap(z,z.^3)
```

例 3-1-20 的结果如图 3-1-20 所示。

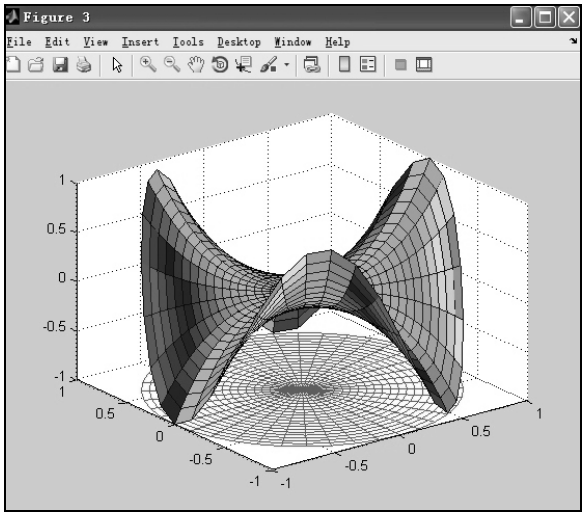


图 3-1-20　绘复变函数图

对于三维图形，通常可以利用 $z=f(x,y)$的确定或不确定的函数关系来绘制可视化图形，此时自变量是二维的；而在高等物理、力学等的研究当中经常会遇到 $v=v(x,y,z)$的函数。此时自变量是三维的，图形应当是四维的。但是由于人所处空间和思维的局限性，在计算机

的屏幕上只能表现出三个空间变量。为了表现四维图像，引入了三维实体的四维切片色图，它由函数 slice 来实现，其调用格式如下：

slice(X,Y,Z,V,Sx,Sy,Sz)：绘制向量 Sx, Sy, Sz 中的点沿 x,y,z 方向的切片图。数组 X,Y,Z 用来定义 V 的坐标。在每一点的颜色必须由对容量 V 的插值来决定。V 必须是 M×N ×P 阶的矩阵。

【例 3-1-21】可视化函数 $f = xe^{-x^2-y^2-z^2}$，自变量的变化范围分别为$-2<x<2$，$-2<y<2$，$-2<z<2$。

程序如下：

```
[x,y,z] = meshgrid(-2:.2:2, -2:.25:2, -2:.16:2);
v = x .* exp(-x.^2 - y.^2 - z.^2);
slice(x,y,z,v,[-1.2 .8 2],2,[-2 -.2])
```

例 3-1-21 的结果如图 3-1-21 所示。

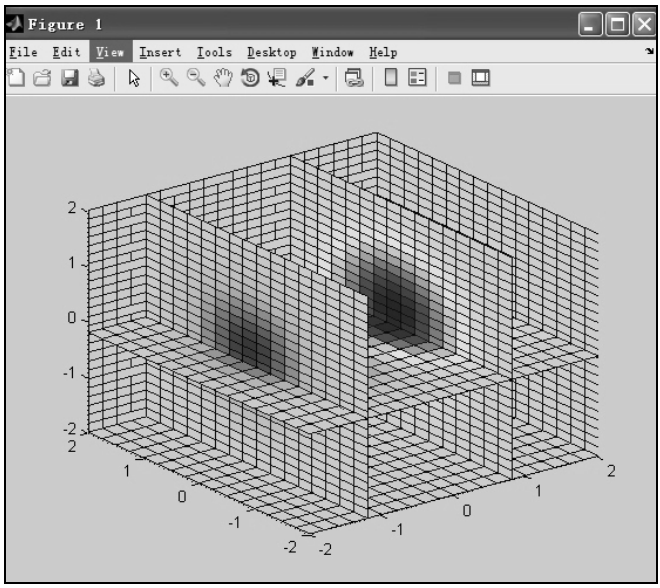


图 3-1-21　函数的四维图像

【例 3-1-22】模拟杨氏双缝干涉实验。

程序如下：

```
clear all
lambda1=[500e-9 700e-9 500e-9 500e-9];     %波长大小
a1=[1e-3 1e-3 10e-3 1e-3] ;                %两狭缝之间距离
D1=[1 1 1 10];                             %狭缝到屏幕的距离
for j=1:4
    lambda=lambda1(j);a=a1(j);D=D1(j);
    ymax=3*lambda*D/a;
    xs=ymax;ny=101;
    ys=linspace(-ymax,ymax,ny);
```

```
        for i=1:ny
            r1=sqrt((ys(i)-a/2).^2+D^2);
            r2=sqrt((ys(i)+a/2).^2+D^2);
            phi=2*pi*(r2-r1)/lambda;
            I(i,:)=4*cos(phi/2).^2;
        end
        nclevels=255;
        br=(I/3.0)*nclevels;                %定标：使最大光强(3.0)对应于最大灰度级
        subplot(2,4,2*j-1),image(xs,ys,br)
        colormap(gray(nclevels));
        subplot(2,4,2*j),plot(I,ys);
    end
```

例 3-1-22 的结果如图 3-1-22 所示。

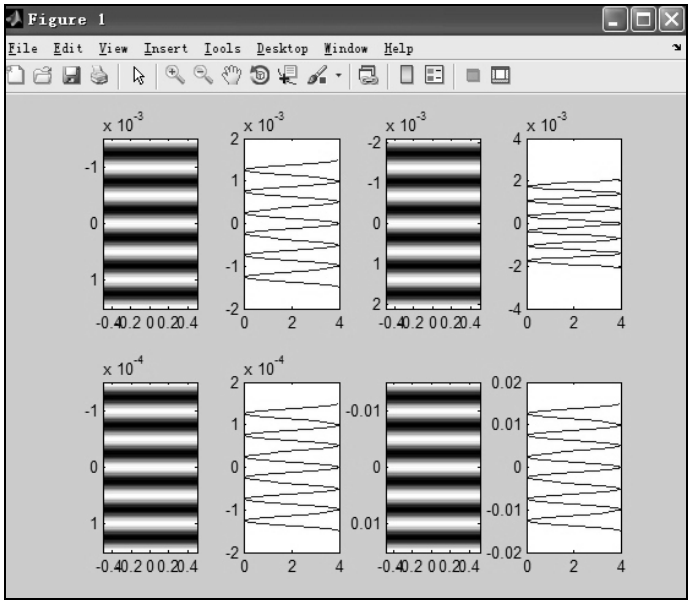


图 3-1-22　模拟杨氏双缝干涉实验

3.1.4　动画制作

MATLAB 制作动画有多种方式：

- 以质点运动轨迹来显示动画；
- 以电影播放的方式来显示动态效果；
- 以对象方式显示。

1．质点运动轨迹

使用 comet、comet3 函数，如 comet(x,y) 显示质点绕向量 y 与 x 运动。

【例 3-1-23】模拟匀速圆周运动。

程序如下：

```
sita = 0:0.0001:2*pi;
r = 10;
x=r*cos(sita);
y=r*sin(sita);
comet(x,y)
```

2．电影播放

电影播放方式首先保存想要产生动画的图片，存储为一系列各种类型的二维、三维图，然后像放电影的方式按次序播放出来。

动画生成的步骤：

（1）调用 moviein 函数对内存进行初始化，创建一个足够大的矩阵，存放当前坐标轴大小的一系列指定图形。

（2）调用 getframe 函数对动画中的每一帧生成图形，并把它们放到帧矩阵中。

（3）调用 movie 函数从帧矩阵中回放动画，可以指定次数和播放速度。

【例 3-1-24】模拟眼球动画。

程序如下：

```
axis equal          %坐标相同
M=moviein(8)        %产生矩阵
set(gca,'Nextplot','replacechildren')
for  j=1:8
   plot(fft(eye(j+8)))
   %eye 为单位矩阵，fft 为快速傅里叶变换
   M(:,j)=getframe;
end
movie(M,2,1)        %M 为播放对象，2 为播放次数，1 为每秒播放的帧数，默认为 12
```

【例 3-1-25】图形放大。

程序如下：

```
[x,y] = meshgrid([-1.05:0.2:75]);
z = x.*exp(-x.^2-y.^2);
axis tight;   %axis limits to the range of the data
set(gca,'nextplot','replacechildren');
for j = 1:40
   surf(x*sin(pi*j/100),y*sin(pi*j/100),z*sin(-pi*j/100));
   m(j) = getframe
end
movie(m)
```

【例 3-1-26】小球绕一曲线前进。

程序如下：

```
clc;clear;
n=100;
```

```
x=0:pi/n:2*pi
y=sin(x);
k=0;
for t=0:pi/n:2*pi
    k=k+1;
    x(k)=t
    y(k)=sin(t);
    m=plot(x,y,x(k),y(k),'or')
    grid
    getframe;
end
```

3．以对象方式显示

设置对象的属性 EraseMode，更新对象来产生新图，drawnow 函数进而覆盖旧图，从而使得图形不断发生变化。在程序循环中，通过改变对象的坐标来移动对象。

【例 3-1-27】旋转红点。

程序如下：

```
x = -pi:pi/30:pi;
h = plot(x,cos(x),'o','MarkerEdgeColor','k','MarkerFaceColor','r',
     'MarkerSize',8,'EraseMode','Xor')
for j = 1:10000
    y = 1/2*sin(3*x+0.006*j);
    set(h,'ydata',y);
    drawnow;
end
```

3.2 图形句柄

句柄图形是利用底层绘图函数，通过对对象属性的设置（Handle Graphics）与操作实现绘图。句柄图形中所有图形操作都是针对图形对象而言的，体现了面向对象的程序设计。

高层绘图与底层绘图的区别在于：

（1）高层绘图函数是对整个图形进行操作的，图形每一部分的属性都是按默认方式设置的，充分体现了 MATLAB 语言的实用性。

（2）底层绘图函数可以定制图形，对图形的每一部分进行控制，用户可以用来开发用户界面以及各专业的专用图形，充分体现了 MATLAB 语言的开发性。

3.2.1 图形对象和句柄

MATLAB 的图形对象包括计算机屏幕、图形窗口、坐标轴、用户菜单、用户控件、曲线、曲面、文字、图像、光源、区域块和方框等。系统将每一个对象按树状结构组织起来。每个具体图形不必包含每个对象，但每个图形必须具备根屏幕和图形窗口，图像对象如图 3-2-1 所示。

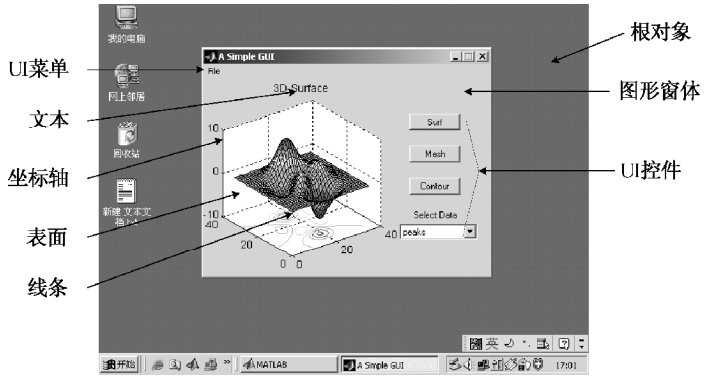


图 3-2-1　图形对象

图形对象之间的关系为父代与子代的关系，如图 3-2-2 所示。

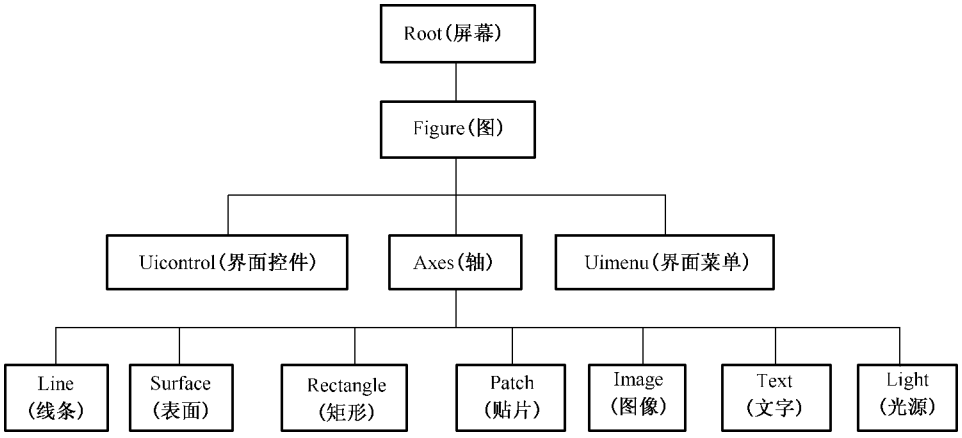


图 3-2-2　图形对象继承关系

MATLAB 在创建每一个图形对象时，都为该对象分配唯一的一个值，称其为图形对象的句柄（Handle）。句柄是图形对象的唯一标识符，不同对象的句柄不能重复和混淆。

计算机屏幕作为根对象由系统自动建立，其句柄值为 0，而图形窗口对象的句柄值为一正整数，并显示在该窗口的标题栏，其他图形对象的句柄为浮点数。MATLAB 提供了若干个函数用于获取已有图形对象的句柄，如 figure，line，text，surface，axes（xlabel，ylabel，zlabel，title）等，这些函数的返回值均为唯一的一个值。

建立图形对象后，可用以下函数对图形句柄进行操作。图形句柄操作函数如表 3-2-1 所示。

表 3-2-1　图形句柄操作函数

函　数	说　　明	函　数	说　　明
findobj	按照指定的属性来获取图形对象的句柄	gcf	获取当前的图形窗口句柄
gca	获取当前的轴对象句柄	gco	获取当前的图形对象句柄
get	获取当前的句柄属性和属性值	set	设置当前句柄的属性值

MATLAB 为每个图形窗口提供了很多属性，这些属性及其取值控制着图形窗口对象。

常用属性有 menuBar 属性、Name 属性、NumberTitle 属性、Resize 属性、Position 属性、Units 属性、Color 属性、Pointer 属性、WindowButtonDownFcn 等。

【例 3-2-1】绘制二维曲线，通过选择不同的选项设置曲线的颜色、线型和数据点的标记符号。

程序如下：

```
x=0:0.1:10;
y=sin(x);
h_f=figure('Position',[200 300 300 300],'menubar','none');
h_a1=axes('position',[0.1,0.1,.8,.8]);
h_t=title(h_a1,'正弦曲线');                    %创建标题
h_l=line(x,y);                                 %设置坐标轴刻度
%设置坐标轴刻度标注
set(gca,'xtick',[0 pi/2 pi 3*pi/2 2*pi 5*pi/2 3*pi])
set(gca,'xticklabel',{'0','pi/2','pi','3*pi/2','2pi','5*pi/2','3pi'})
set(gca,'xgrid','on','ygrid','on');            %设置坐标轴属性
set(h_l,'linewidth',2)                         %设置线属性
set(get(h_t,'parent'),'color','y')             %设置标题的父对象属性
%创建矩形框
h_ann0=annotation(gcf,'rectangle',[0.1 0.5 .8 0.4],'FaceAlpha',.7,
                  'FaceColor','red');
```

例 3-2-1 的结果如图 3-2-3 所示。

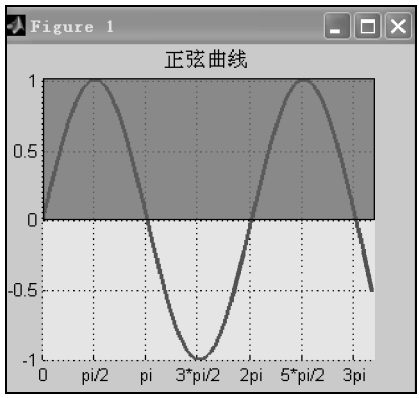


图 3-2-3　句柄方式绘制正弦函数

【例 3-2-2】利用坐标轴对象实现图形窗口的任意分割。

程序如下：

```
clf
x=0:pi/10:2*pi
y=sin(x);
axes('position',[0.2,0.2,0.2,0.7]);
plot(y,x);
grid on
```

```
set(gca,'gridlinestyle','--');
axes('position',[0.4,0.2,0.5,0.5]);
t=0:pi/100:20*pi
x=sin(t);
y=cos(t);
z=t.*sin(t).*cos(t);
plot3(x,y,z)
axes('position',[0.55,0.6,0.25,0.3]);
[X,Y]=meshgrid(-8:0.5:8)
Z=sin(sqrt(X.^2+Y.^2))./sqrt(X.^2+Y.^2+eps);
mesh(X,Y,Z)
```

例 3-2-2 的结果如图 3-2-4 所示。

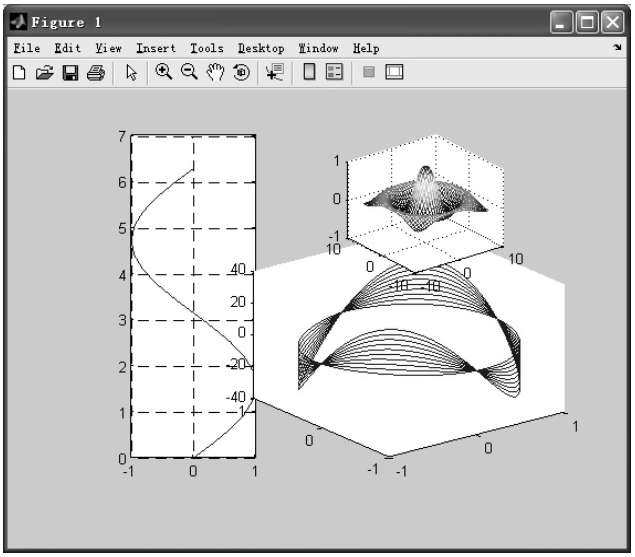


图 3-2-4　坐标轴对象的任意分割

【例 3-2-3】利用曲面对象绘制三维曲面 $z=\sin(x)$。

程序如下：

```
x=0:pi/20:2*pi;
[X,Y]=meshgrid(x);
Z=sin(X);
axes('view',[-37.5,80])
hs=surface(X,Y,Z,'facecolor','g','edgecolor','r');
grid on
set(get(gca,'xlabel'),'string','x-axis');
set(get(gca,'ylabel'),'string','y-axis');
set(get(gca,'zlabel'),'string','z-axis');
title('mesh-surf')
```

例 3-2-3 的结果如图 3-2-5 所示。

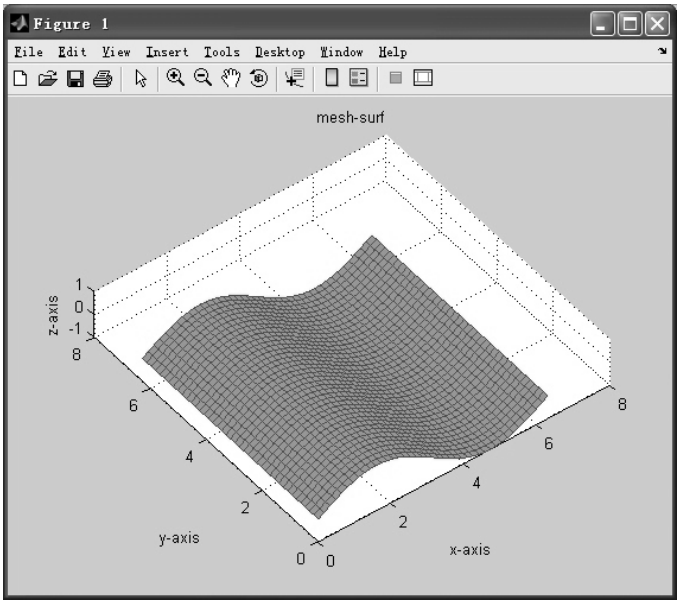


图 3-2-5　三维曲面

利用句柄图形还可绘制用户界面菜单和快捷菜单。

快捷菜单是用鼠标右键单击某对象时在屏幕上弹出的菜单，这种菜单出现的位置是不固定的，而且总是和某个图形对象相联系的。在 MATLAB 中，可以使用 uicontextmenu 函数和图形对象的 UIContextmenu 属性来建立快捷菜单，具体步骤为：

（1）利用 uicontextmenu 函数建立快捷菜单。

（2）利用 uimenu 函数为快捷菜单建立菜单项。

（3）利用 set 函数将该快捷菜单和某图形对象联系起来。

【例 3-2-4】绘制曲线 $y=2\sin(5x)\sin x$，并建立一个与之相联系的快捷菜单，用以控制曲线的线型和曲线宽度。

程序如下：

```
x=0:pi/100:2*pi
y=2*sin(5*x).*sin(x);
h1=plot(x,y);
hc=uicontextmenu
hls=uimenu(hc,'label','线型')
hlw=uimenu(hc,'label','线宽')
uimenu(hls,'label','虚线','callback','set(h1,''linestyle'','':'');');
uimenu(hls,'label','实线','callback','set(h1,''linestyle'',''-'');');
uimenu(hlw,'label','加宽','callback','set(h1,''linewidth'',5);');
uimenu(hlw,'label','变细','callback','set(h1,''linewidth'',1);');
set(h1,'uicontextmenu',hc)
```

例 3-2-4 的结果如图 3-2-6 所示。

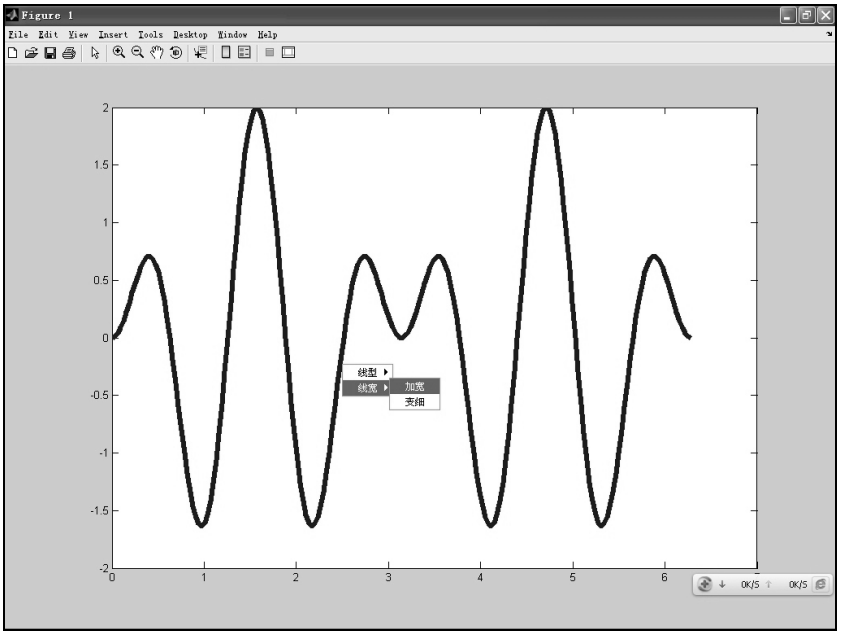


图 3-2-6　带快捷菜单的函数绘制

3.2.2　利用句柄图形设计 GUI

GUI 即图形用户界面，是指由窗口、菜单、图标、光标、按键、对话框和文本等各种图形对象组成的用户界面，它可让用户定制与 MATLAB 的交互方式。

【例 3-2-5】建立数制转换对话框。在左边输入一个十进制整数和 2～16 之间的数，单击“转换”按钮能在右边得到十进制数所对应的二～十六进制字符串，单击“退出”按钮退出对话框。

程序如下：

```
hf=figure('Color',[0,1,1],'Position',[100,200,400,200],...
'Name','数制转换','NumberTitle','off','menuBar','none');
uicontrol(hf,'Style','Text','Units','normalized',...
'Position',[0.05,0.8,0.45,0.1],'Horizontal','center',...
'String','输入框','Back',[0,1,1]);
uicontrol(hf,'Style','Text','Position',[0.5,0.8,0.45,0.1],...
'Units','normalized','Horizontal','center',...
'String','输出框','Back',[0,1,1]);
uicontrol(hf,'Style','Frame','Position',[0.04,0.33,0.45,0.45],...
'Units','normalized','Back',[1,1,0]);
uicontrol(hf,'Style','Text','Position',[0.05,0.6,0.25,0.1],...
'Units','normalized','Horizontal','center',...
'String','十进制数','Back',[1,1,0]);
uicontrol(hf,'Style','Text','Position',[0.05,0.4,0.25,0.1],...
'Units','normalized','Horizontal','center',...
'String','2～16 进制','Back',[1,1,0]);
```

```
he1=uicontrol(hf,'Style','Edit','Position',[0.25,0.6,0.2,0.1],...
'Units','normalized','Back',[0,1,0]);
he2=uicontrol(hf,'Style','Edit','Position',[0.25,0.4,0.2,0.1],...
'Units','normalized','Back',[0,1,0]);
uicontrol(hf,'Style','Frame','Position',[0.52,0.33,0.45,0.45],...
'Units','normalized','Back',[1,1,0]);
ht=uicontrol(hf,'Style','Text','Position',[0.6,0.5,0.3,0.1],...
'Units','normalized','Horizontal','center','Back',[0,1,0]);
COMM=['n=str2num(get(he1,''String''));','b=str2num(get(he2,''String''));'
,...
'dec=trdec(n,b);','set(ht,''string'',dec);'];
uicontrol(hf,'Style','Push','Position',[0.18,0.1,0.2,0.12],...
'String','转 换','Units','normalized','Call',COMM);
uicontrol(hf,'Style','Push','Position',[0.65,0.1,0.2,0.12],...
'String','退 出','Units','normalized','Call','close(hf)');
```

程序调用了 trdec.m 函数文件，该函数的作用是将任意十进制整数转换为二～十六进制字符串。trdec.m 函数文件如下：

```
function dec=trdec(n,b)
ch1='0123456789ABCDEF';          %十六进制的 16 个符号
k=1;
while n~=0                       %不断除某进制基数取余直到商为 0
    p(k)=rem(n,b);
    n=fix(n/b);
    k=k+1;
end
k=k-1;
strdec='';
while k>=1                       %形成某进制数的字符串
    kb=p(k);
    strdec=strcat(strdec,ch1(kb+1:kb+1));
    k=k-1;
end
dec=strdec;
```

例 3-2-5 的结果如图 3-2-7 所示。

图 3-2-7　数值转换 GUI

3.3 GUI 设计工具 GUIDE

使用图形句柄创建 GUI 的过程繁琐，为了便于创建图形用户界面，MATLAB 提供了一个开发环境，能够帮助用户创建图形用户界面，这就是 GUIDE（Graphic User Interface Development Environment），如图 3-3-1 所示。

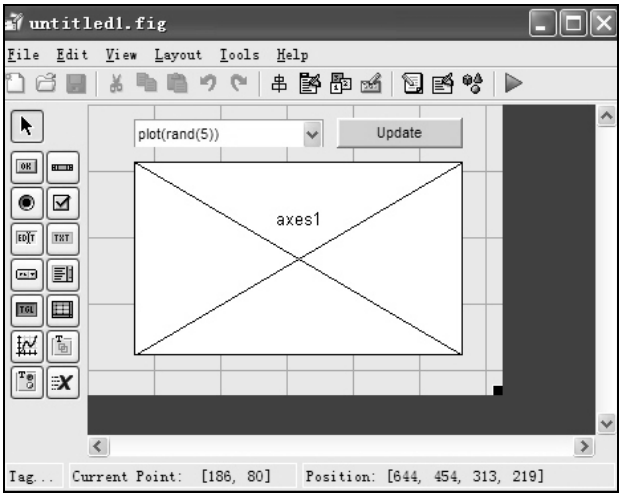


图 3-3-1 GUIDE 集成环境

在 MATLAB 中可通过输入 GUIDE 指令或通过“Start”菜单选择“MATLAB”下的“GUIDE”命令等多种方式进入操作界面。GUIDE 提供多个模板来定制 GUI，这些模板均已包括相关的回调函数，可以通过修改对应的 M 文件函数，实现指定功能。

在模板设计界面中，可以选择创建新的 GUI 或者打开原有的 GUI。在创建新的 GUI 时 MATLAB 提供以下 4 种模板：

- 空白模板；
- 带有控制按钮模板；
- 带有坐标轴和菜单模板；
- 问答式对话框模板。

类似于目前流行的各种面向对象开发语言，GUIDE 界面中包含一些集成环境供用户使用，主要有

- 对象浏览器（Object Browser）：用于获得当前 MATLAB 图形用户界面程序中的全部对象信息，对象的类型，同时显示控件的名称和标识，在控件上双击鼠标可以打开该控件的属性编辑器；
- 布局编辑器（Layout Edtor）：在图形窗口中创建及布置图形对象；
- 几何排列工具（Alignment Tool）：调整各对象相互之间的几何关系和位置；
- 属性查看器（Property Inspector）：查询并设置属性值；
- 菜单编辑器（menu Editor）：创建、设计、修改下拉式菜单和快捷菜单；

- Tab 顺序编辑器（Tab Order Editor）：设置当用户按下键盘上的 Tab 键时，对象被选中的先后顺序；
- 利用 GUIDE 创建的 GUI 包含两个文件：M 文件和外观 fig 文件。

一般来说，使用 GUIDE 设计一个 GUI 的应用程序的步骤如下。

（1）界面布局设计。界面布局设计包括以下几个步骤：

① 通过拖拽控件面板中的控件到界面设计区中；

② 使用对象对齐工具（Align Objects）进行控件的布局调整，使用 Tab 顺序编辑器（Tab Order Editor）对各控件的 Tab 顺序进行较好地设置；

③ 如果界面需要菜单，则使用菜单编辑器（Menu Editor）进行菜单的设计；

④ 添加完控件后在对象浏览器（Object Browser）中即可看到所有的图形对象，完成界面的布局设计。

（2）属性设置。打开属性编辑器（Property Inspector）对相关的属性进行修改 。

（3）编写回调函数。

回调函数的声明为

```
function object_Callback(hObject, eventData, handles)
```

其中，object：发生事件的控件的 Tag 属性字符串；hObject：发生事件的控件的句柄；eventData：保留字段，目前版本的 MATLAB 还暂时不使用；handles：一个结构，包含所有界面上控件的 Tag 属性值，可添加用户自己的数据。

函数的头部为程序的初始化和调度代码，一般情况下，用户不需要修改这部分代码。在程序执行的过程中，这部分代码起到了调度程序的功能，分别完成了打开图形界面、初始化以及响应用户动作的功能。在调度代码的后面紧跟着两个子函数，这两个子函数就是 GUI 的回调函数。第一个回调函数是

```
function simple_gui_OpeningFcn(hObject, eventdata, handles, varargin)
```

该函数负责打开图形界面，同时，若程序中需要对一些全局的参数进行初始化或者设置时，可以将初始化用户数据的代码添加在该子函数中。

第二个回调函数是

```
function varargout =simple_gui_OutputFcn(hObject, eventdata, handles)
```

该子函数负责将图形界面的句柄返回给用户的输出参数。

接下来的子函数是分别用来响应用户的动作输入，完成相应功能的 GUI 控件回调子函数。

注意：在 GUIDE 创建的 M 函数文件中，若修改了 handles 结构，则需要通过 guidata 函数将 handles 的结构保存起来，只有这样才能够通过 handles 结构将不同的用户数据传递到相应的子函数中。

对于例 3-2-5，若使用 GUIDE 设计，可通过工具设计界面及各控件的外观，然后将 trdec.m 函数内容放入转换按钮所对应的回调函数处，按格式进行适当调整即可。

第 4 章　数字图像处理

图像信息是获得外界信息的主要来源之一，在科学研究、军事技术、工农业生产、医学、气象及天文学等领域中，人们越来越多地利用图像信息来认识和判断事物，解决实际问题。图像处理技术的研究内容涉及光学系统、微电子技术、计算机科学、数学分析等领域，从某种意义上来说，图像处理也是一种对实际图像进行仿真的技术。MATLAB 是基于矩阵的高级程序语言，而数字图像实际上就是一组有序的离散数据，从而 MATLAB 从本质上就提供了对图像处理的技术支持。

4.1　图像处理概述

4.1.1　基本概念

1．图像

图像是各种图形和影像的总称，是二维或三维的空间信息，包含事物的形态、位置和色彩信息等，以便人们进行观察、测量和识别。图像包括静止图像和运动图像。

组成数字图像的基本单位是像素，数字图像是像素的集合。像素为元素的矩阵，像素的值代表图像在该位置的亮度，称为图像的灰度值。像素具有整数坐标和整数灰度值。

用来描述图像的属性有以下几种。

（1）亮度：也称为灰度，它是颜色的明暗变化，常用 0%～100%（由黑到白）表示。

（2）对比度：是画面黑与白的比值，即从黑到白的渐变层次。比值越大，从黑到白的渐变层次就越多，色彩表现越丰富。

（3）直方图：图像在计算机中的存储形式，就是由像素点组成的矩阵，每个点上的值就是图像的灰度值，直方图就是每种灰度在这个点矩阵中出现的次数。反映图像中每种灰度出现的频率。

2．图像的类型及存储

计算机中，数字图像的存储都是矩阵方式的。矩阵中的每个点代表图像对应位置的像素。MATLAB 中，图像的类型及存储格式有如下几种。

（1）二值图像：显示黑、白信息的图像。只需一个数据矩阵，每个像素只取 0、1 两个灰度值。

（2）灰度图像：是包含灰度级（量度）的图像。仅包含一个数据矩阵，其中数据矩阵中的每一个数据代表了一定范围内的灰度值（0～255）。

（3）索引图像：直接把像素值作为 RGB 调色板下标的图像。包括一个数据矩阵及一个颜色映像矩阵。

（4）RGB 图像：即真彩图像，图像的数据不仅包含亮度信息，还要包含颜色信息。用 RGB 三原色表示图像色彩信息。每一个像素值由三个数值来指定红、绿和蓝颜色分量，如要读取图像中（100，50）处的像素值，可查看三元数据（100，50，1:3）。

（5）多帧图像阵列：由多帧图像组成，每一帧图像可以为前 4 种图像中的一种，但组成一个多帧图像阵列的图像必须为同一种。

不同的操作系统和图像处理软件所支持的图像格式不同。常见的图像格式如表 4-1-1 所示。

表 4-1-1　常见的图像格式

文　　件	特点及用途
bmp	Windows 标准的点阵式图形文件格式，图像信息比较丰富，但占用磁盘空间较大
gif	用于动画、多媒体程序界面，网页界面，占用磁盘空间较小
tif	用于专业印刷，采用无损压缩方式
jpg	用于数字图片保存、传送，是目前压缩率最高的格式

3．图像的变换

数字图像处理可在空间域和频域中进行。图像本身所在的域称为空间域，在空间域中处理图像，包括各种统计方法、微分方法及其他数学方法。图像灰度值随空间坐标变化的快慢可用频率来度量，称为空间频率（Spatial Frequency）。频域处理包括各种正交变换和图像滤波等方法，其共同点是将图像变换到频域中进行处理后，再变换到原来的空间域中。

每一种变换都有自己的正交函数集，引入不同的变换，常见的变换有：

- 傅里叶变换；
- 余弦变换；
- 正弦变换；
- 沃尔什变换；
- 小波变换。

4．图像处理

图像处理一般就是指数字图像处理，是将图像信号转换成数字信号并利用计算机对其进行分析、加工和处理，使其满足视觉、心理和模式识别技术的需要。

早期的图像处理的目的是改善图像的质量，以改善人的视觉效果为目的。图像处理中，输入的是质量低的图像，输出的是改善质量后的图像，常用的图像处理方法有图像增强、复原、编码、压缩等。随着计算机技术和人工智能、思维科学研究的迅速发展，数字图像处理已开始研究如何用计算机系统解释图像，实现类似人类视觉系统理解外部世界，即图像理解或计算机视觉。

从图像处理学的角度，研究一幅图像，可以从三个层次去理解，各层次的主要内容有：

（1）底层信息，包括

- 像素：每幅图像存储为矩阵[$a(i,j)$]，其中(i, j)为像素空间位置，$a(i,j)$为像素的（红/绿/蓝）三通道颜色向量或者灰度值。
- 边缘：颜色或者灰度变化比较大的像素点集合。生物视觉系统对边缘具有强敏感性。
- 纹理：草地、树林、天空等区域。

（2）中层信息，包括

- 分割：根据图像的特点和用户的需求可分割为不同的区域。
- 目标检测：例如利用算法自动识别与标记图像中的不同物体。

（3）高层信息。图像语义描述，由图像中所包含物体的特征所推理得到的语义描述，高层语义描述具有不唯一性。

4.1.2 图像的运算

数字图像的基本单元是像素，用计算机对图像进行处理有多种运算方式，基本的运算形式有：

（1）点运算。在对图像各像素进行处理时，只输入该像素本身灰度的运算方式称为点运算。对图像作点运算时各像素间不发生关系，各像素的处理是独立进行的。点运算最常用的一种应用就是直方图的均衡化。

（2）邻域运算。在对图像各像素进行处理时，不仅输入该像素本身灰度，还要输入以该像素为中心的某局部区域（即邻域）的一些像素进行运算的方式，称为邻域运算。邻域运算能将周围邻域内的诸像素状况反映在处理结果中，因而便于实现多种处理内容。

（3）并行运算。并行运算指的是对图像上各像素同时进行相同处理的运算方式。这种运算方式处理速度快，但只能用于处理结果与处理顺序无关的场合。点运算处理可采用并行运算方式。对于邻域运算的处理能否采用并行方式不能一概而论。

（4）串行运算。串行运算指的是在图像上按照规定的顺序逐个对像素进行处理的运算形式。凡是对像素的处理在邻域像素的基础上进行的处理方法，都必须采用串行运算形式，并同时规定处理顺序。点运算可以采用串行运算。

（5）迭代运算。反复多次进行相同处理的运算，称为迭代运算。迭代运算常用于一次运算不能达到处理目的的情况。

（6）窗口运算。窗口运算是指对图像特定的矩形区域进行某种运算的形式。一般，矩形区域由矩形左上角 S 点的坐标(a,b)和矩形所包含的行数 m 和列数 n 确定。矩形区域可以是图像中存在某对象物的位置，也可以是图像中具有代表性特征的区域。

（7）模板运算。对图像中特定形状的区域进行某种运算的方式称为模板运算。这里的模板是指特定形状的区域，它常常是与图像中存在的对象有相同特征的一个局部的子图像，因此模板实际上就是一个二维数组。通过对图像上各像素的模板运算，可以找到图像上与模板特征相同的对象物的存在位置。

（8）帧运算。通常一幅完整的图像被称为一帧，在两幅或多幅图像之间进行运算产生一幅新图像的处理称之为帧运算。帧运算可看成一种图像合成处理，在运算时，将两幅或多幅图像中的对应点用位逻辑运算或算术运算方法进行合成。

4.1.3 图像处理的内容

1．图像处理的特点

图像处理具有以下特点：

（1）处理信息量大。数字图像处理的信息大多是二维信息，处理信息量很大。例如，一幅彩色 512×512 图像，要求 768 KB；如果要处理 30 帧/秒的电视图像序列，则每秒要求 500 KB～2.5 MB 的数据量，对计算机的计算速度、存储容量等要求较高。

（2）占用的频带较宽。与音频信息相比，图像占用的频带要大 n 个数量级。例如，电视图像的带宽约为 5.6 MHz，而语音带宽仅为 4 kHz 左右，所以在成像、传输、存储、处理、显示等各个环节的实现上，难度较大，成本高，这就对频带压缩技术提出了更高的要求。

（3）复现三维景物的能力差。由于图像是三维景物的二维投影，一幅图像本身不具备复现三维景物的全部几何的能力，三维景物背后部分信息在二维图像画面上是反映不出来的，因此，要理解三维景物必须作合适的假定或附加新的测量。这也是人工智能正在致力解决的知识工程问题。

（4）人为因素影响大。经处理后的图像一般是给人观察和评价的，因此受人的因素影响较大。由于人的视觉系统很复杂，受环境条件、视觉性能、情绪爱好以及知识状况影响很大，作为图像质量的评价还有待进一步深入的研究。另一方面，计算机视觉是模仿人的视觉，人的感知机理必然影响着计算机视觉的研究。

2．图像处理的方法

具体来讲，图像处理研究包含的主要内容有：

（1）图像变换。由于图像阵列很大，直接在空间域中进行处理，涉及计算量很大。因此，往往采用各种图像变换的方法，如傅里叶变换、沃尔什变换、离散余弦变换等间接处理技术，将空间域的处理转换为变换域处理，不仅可减少计算量，而且可获得更有效的处理（如傅里叶变换可在频域中进行数字滤波处理）。

（2）图像增强和复原。图像增强和复原的目的是为了提高图像的质量，如去除噪声、提高图像的清晰度等。图像增强不考虑图像降质的原因，突出图像中所感兴趣的部分，如强化图像高频分量，可使图像中物体轮廓清晰，细节明显；强化低频分量可减少图像中噪声影响。图像复原要求对图像降质的原因有一定的了解，一般应根据降质过程建立“降质模型”，再采用某种滤波方法，恢复或重建原来的图像。

（3）图像编码压缩。图像压缩主要目的是为了节省存储空间，增加传输速度。图像压缩的理想标准是信息丢失最少，压缩比例最大。不损失图像质量的压缩称为无损压缩，无损压缩不可能达到很高的压缩比；损失图像质量的压缩称为有损压缩，高的压缩比是以牺牲图像质量为代价的。压缩的实现方法是对图像重新进行编码，希望用更少的数据表示图像。

信息的冗余量有许多种，如空间冗余、时间冗余、结构冗余、知识冗余、视觉冗余等，数据压缩实质上是减少这些冗余量。高效编码的主要方法是尽可能去除图像中的冗余成分从而以最小的码元包含最大的图像信息。

（4）图像分割。图像分割是将图像中有意义的特征部分提取出来，有意义的特征包括图像中的边缘、区域等，这是进一步进行图像识别、分析和理解的基础。虽然目前已研究出不少边缘提取、区域分割的方法，但还没有一种普遍适用于各种图像的有效方法，是目前图像处理中研究的热点之一。

（5）图像描述。图像描述是图像识别和理解的必要前提。作为最简单的二值图像可采用其几何特性描述物体的特性，一般图像的描述方法采用二维形状描述，它有边界描述和区域描述两类方法。对于特殊的纹理图像可采用二维纹理特征描述。随着图像处理研究的深入发展，已经开始进行三维物体描述的研究，提出了体积描述、表面描述、广义圆柱体描述等方法。

（6）图像分类（识别）。图像分类（识别）属于模式识别的范畴，其主要内容是图像经过某些预处理（增强、复原、压缩）后，进行图像分割和特征提取，从而进行判决分类。图像分类常采用经典的模式识别方法，有统计模式分类和句法（结构）模式分类，近年来新发展起来的模糊模式识别和人工神经网络模式分类在图像识别中也越来越受到重视。

3．图像处理的应用

图像处理目前应用的领域非常广泛，如

- 遥感图像应用：资源调查、灾害监测、农林业规划、城市规划、环境保护等；
- 医学图像应用：计算机断层摄影计算成像 CT 技术、X 射线、染色体分析等；
- 工业和实验图像应用：无损探伤、自动检查和识别、智能机器人等；
- 办公室自动化图像应用：邮政编码图像识别、OCR（字符识别系统）、自动判卷系统、各类图纸自动识别与录入系统等；
- 军事公安图像应用：自动跟踪技术、指纹识别、不完整图片的复原、监控等；
- 文化艺术图像应用：服装设计、照片的复制和修复、运动员动作分析等；
- 图像数据传输应用：图像的存储、刻盘、互联网传输、卫星传输、无线传输等。

4.2 图像处理的应用

4.2.1 图像处理工具箱

MATLAB 对图像的处理功能主要集中在它的图像处理工具箱（Image Processing Toolbox）中。图像处理工具箱是由一系列支持图像处理操作的函数组成的，可以进行诸如几何操作、线性滤波和滤波器设计、图像变换、图像分析与图像增强、二值图像操作以及形态学处理等图像处理操作。工具箱中大部分函数均以开放式 MATLAB 语言编写，用户可以检查算法、修改源代码和创建自定义函数。工具箱中还包括两个小工具：Image Tool 能显示一幅图像和图像的基本信息，如存储尺寸，颜色类型，编码方法等；Movie Player 能播放视频和显示视频的基本信息。

MATLAB 支持五种图像类型，即索引图像、灰度图像、二值图像、RGB 图像和多帧

图像阵列；支持 BMP、GIF、HDF、JPEG、PCX、PNG、TIFF、XWD、CUR、ICO 等图像文件格式的读、写和显示。

基本的图像处理操作一般包括读入图像、显示图像、处理图像和存储图像等几个部分，使用 MATLAB 函数完成上述操作通常只需非常简单的几行代码即可实现，相比于 C 语言等其他常用图像处理语言，MATLAB 大大减轻了图像数据的繁杂操作，使用户不必花很多的时间在代码的调试上，而是把更多的精力放于各种图像处理算法的效果上。但由于 MATLAB 的运算速度所限，在一些对时间要求严格的实时处理系统，往往需要使用 MATLAB 和其他工具的结合使用才能达到满意的效果。

图像处理工具箱常用的部分函数见表 4-2-1，应当注意的是很多图像处理函数所针对的图像类型是有指定的，所以调用前需通过转换函数进行类型转换。

表 4-2-1　常用的图像处理函数

函　数	功　能	函　数	功　能
imread	读入并显示一幅图像	imwrite	保存图像
imshow	显示图像	montage	显示多帧图像阵列
immovie	动画显示	subimage	一个图像窗口内使用多个调色板
im2bw	将图像转换为二进制图像	gray2ind	灰度图像转换为索引图像
ind2gray	索引图像转换为灰度图像	rgb2gray	RGB 图像转换为灰度图像
imadd	加法运算	imsubtract	减法运算
immultiply	图形的乘法	imdivide	图像的除法
imresiz	图像大小调整	imrotate	图像旋转
imcrop	图像剪裁	blkproc	显示块从影像中提取
bestblk	选择图像块的尺寸	imadjust	对比度调整
log	对数变换	imhist	直方图显示
histeq	直方图均化	fft	一维傅里叶变换
fft2	二维傅里叶变换	ifft2	二维傅里叶反变换
imnoise	生成模拟噪声	fspecial	产生预定义滤波器
filter2	线性滤波	medfilt2	二维中值滤波器
medfilt2	中值滤波	ordfilt2	状态统计滤波器
edge	边缘检测	wiener2	二维自适应除噪滤波器
imdilate	膨胀运算	imerode	腐蚀运算
roifill	对特定区域进行填充	bwboundarie	连通区域

4.2.2　图像的读写和显示

MATLAB 中，imread 函数用于读入各种图像文件，imwrite 函数用于输出图像，imfinfo 函数用于读取图像文件的有关信息。把图像显示于屏幕有 imshow、image 等函数。subplot 函数能将一个图像窗口分成几个部分，但同一个图像窗口内只能有一个调色板。subimage 函数可在一个图像窗口内使用多个调色板，使得各种图像能在同一个图像窗口中显示，zoom 函数可实现对图像的缩放。

【例 4-2-1】图像的读、写及显示。

程序如下：

```
clear
tu=imread('football.jpg');
imshow(tu);
subplot(1,3,1)
imshow(tu(:,:,[1 2 3]))
subplot(1,3,2)
imshow(tu(:,:,[3 2 1]))
subplot(1,3,3)
imshow(tu(:,:,[1 3 2]))
```

例 4-2-1 的结果如图 4-2-1 所示。

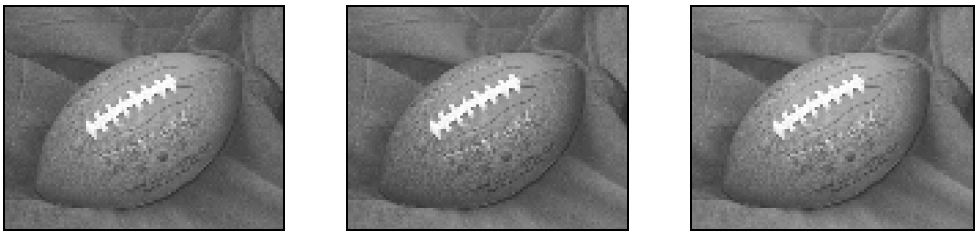


图 4-2-1　图像的读、写及显示

由于图像处理工具箱中很多函数对图像类型有限制，需先用类型转换函数进行转换。例如，要对一幅索引色图像滤波，首先应该将它转换成真彩色图像或者灰度图像，这时 MATLAB 将会对图像的灰度进行滤波，即通常意义上的滤波。如果不将索引色图像进行转换，MATLAB 则对图像调色板的序号进行滤波，这是没有意义的。

【例 4-2-2】图像类型的转换。

程序如下：

```
load trees
BW=im2bw(X,map,0.4);
subplot(1,2,1)
imshow(X,map)
subplot(1,2,2)
imshow(BW)
```

例 4-2-2 的结果如图 4-2-2 所示。

图 4-2-2　图像类型的转换

4.2.3 图像的代数运算

代数运算是指将两幅或多幅图像通过对应像素之间的加、减、乘、除运算得到输出图像的方法。一般地，图像的代数运算是像素对像素的运算，图像代数运算不改变图像的数据存储格式，运算过程中均以浮点数进行运算，运算结果转换为原来的数据类型。

图像的代数运算具有明显的物理意义，例如，图像相加可以将一幅图像的内容加到另一幅图像上，可以实现二次曝光，也可对同一个场景的多幅图像求平均值，这样可以降低噪声。图像相减可以用于运动检测或去除图像中不需要的加性图案。

【例 4-2-3】两幅图像相加。

程序如下：

```
I=imread('rice.png');
J=imread('cameraman.tif');
K=imadd(I,J);
subplot(131);imshow(I);
subplot(132);imshow(J);
subplot(133);imshow(K);
```

例 4-2-3 的结果如图 4-2-3 所示。

图 4-2-3　图像相加

4.2.4 图像的几何处理

图像的几何处理主要包括坐标变换，图像的缩放、旋转、裁剪、扭曲等。

图像的几何运算改变图像的形状，涉及空间变换和灰度插值，空间变换防止图像内容支离破碎，灰度插值计算目标图像中对应原图像非整点的像素灰度值。

【例 4-2-4】图像的放大处理。

程序如下：

```
load woman2
figure
imshow(X,map)
X1=imresize(X,2);
figure
```

```
imshow(X1,[]);
X2=imresize(X,3);
figure
imshow(X2,[]);
X3=imresize(X,4);
figure
imshow(X3,[]);
```

例 4-2-4 的结果如图 4-2-4 所示。

图 4-2-4　图像的放大

【例 4-2-5】图像旋转。

程序如下：

```
[I,map]=imread('kids.tif');
J=imrotate(I,35);
subplot(1,2,1)
imshow(I,map)
subplot(1,2,2)
imshow(J,map)
```

例 4-2-5 的结果如图 4-2-5 所示。

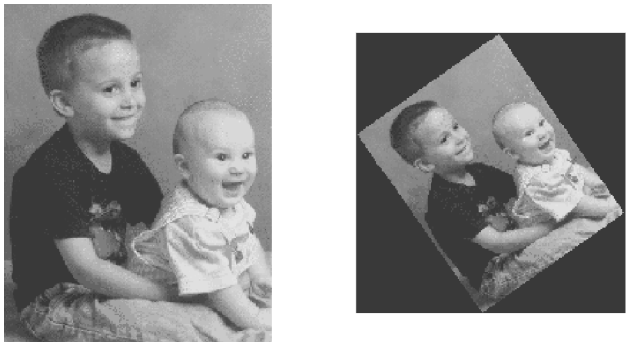


图 4-2-5　图像旋转

4.2.5　图像的增强和复原

图像增强和复原的目的是为了提高图像的质量，常用的图像增强方法有以下几种。

1．直方图均化

灰度直方图用于显示图像的灰度值分布情况，自然图像的灰度直方图通常在低灰度区间上频率较大，使得图像中较暗区域中的细节看不清楚。采用直方图修整可使原图像灰度集中的区域拉开或使灰度分布均匀，从而增大反差，使图像的细节清晰，达到增强目的。直方图均衡化是一种非线性变换，可用 histeq 函数实现。

2．对比度调整

照片或电子方法得到的图像，常表现出低对比度（即整个图像偏亮或偏暗），为此需要对图像中的每一像素的灰度级进行灰度变换，扩大图像灰度范围，以达到改善图像质量的目的。这一灰度调整过程可用 imadjust 函数实现。

3．对数变换

对数变换实现了图像灰度扩展和压缩的功能。它扩展低灰度值而压缩高灰度值，让图像的灰度分布更加符合人的视觉特征。log 函数可用于数字图像的对数变换。

4．图像滤波

图像滤波是指在尽量保留图像细节特征的条件下对目标图像的噪声进行抑制。设计一个适合、匹配的滤波器和恰当的阈值是滤波效果好坏的关键，图像滤波分空域处理和频域处理两类，常用的有均值滤波、中值滤波、高斯滤波、最小均方差滤波、Gabor 滤波等方法。

1）均值滤波

实现图像平滑最常见的方法是在像素邻域内求局部均值，称为均值滤波。

2）中值滤波

中值滤波与均值滤波的区别仅限于：中值滤波是求局部中值而不是局部均值，即对参与计算的像素灰度值按大小排序，然后取位置居中的像素灰度值。

图像处理工具箱里也设计了许多的滤波器，如基于卷积的图像滤波函数 filter、二维卷积滤波函数 conv2 等。

5．图像锐化

图像锐化就是补偿图像的轮廓，增强图像的边缘及灰度跳变的部分，使图像变得清晰，锐化的方法有微分法和高通滤波法等。

【例 4-2-6】直方图均衡及等值化处理。

程序如下：

```
I=imread('cameraman.tif');
subplot(221)
imshow(I)
title('primeval figure ')
J=histeq(I);
subplot(222)
```

```
imhist(I,255)
subplot(223)
imshow(J)
subplot(224)
imhist(J,255)
figure
imcontour(I)
title('image contour')
```

例 4-2-6 的结果如图 4-2-6 和图 4-2-7 所示。

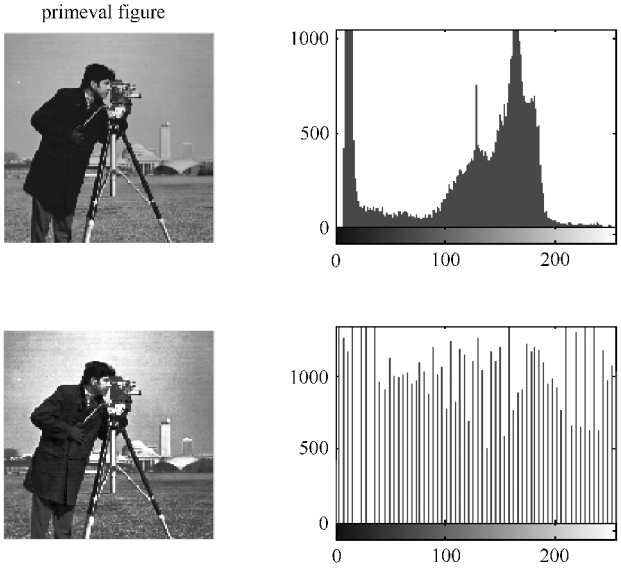


图 4-2-6　直方图均衡

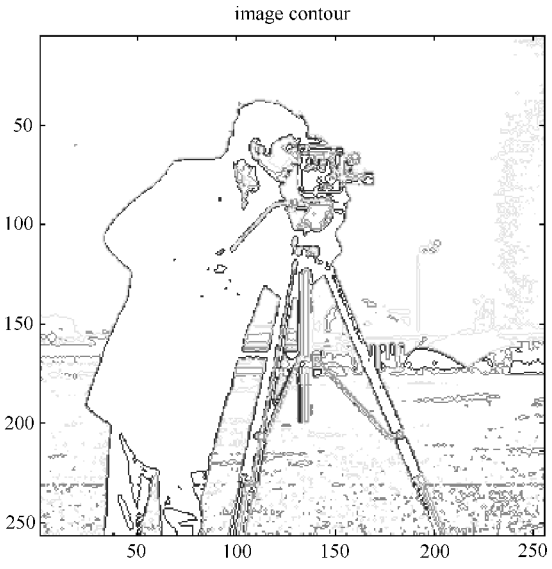


图 4-2-7　等值化图形

【例 4-2-7】对比度调整。

程序如下：

```
I=imread('pout.tif');
subplot(2,2,1);imshow(I);
subplot(2,2,2);imhist(I);
J=imadjust(I,[0.3 0.7],[0 1]);
subplot(2,2,3);imshow(J);
subplot(2,2,4);imhist(J);
```

例 4-2-7 的结果如图 4-2-8 所示。

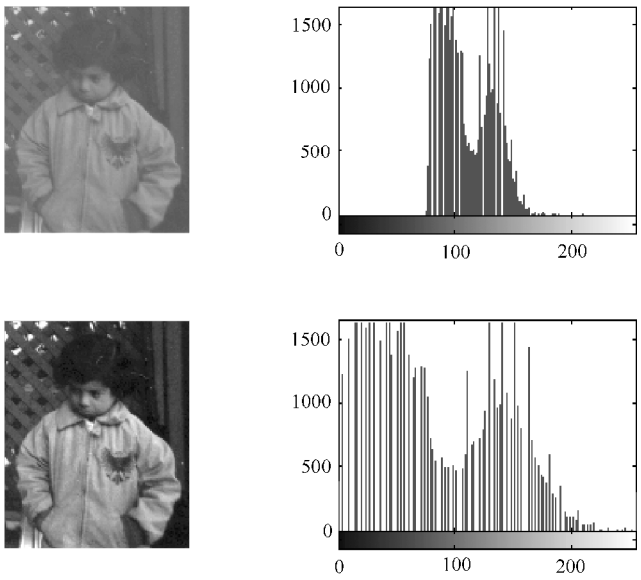


图 4-2-8　对比度调整

【例 4-2-8】不同噪声的叠加。

程序如下：

```
A=imread('dog.jpg');
image(A)
J1=imnoise(A,'gaussian',0,0.2);%这部分数字越大，越模糊，下面的也相似
J2=imnoise(A,'salt & pepper',0.02);
J3=imnoise(A,'speckle',0.02);
subplot(2,2,1),imshow(A),title('原图像');
subplot(2,2,2),imshow(J1),title('加高斯噪声');
subplot(2,2,3),imshow(J2),title('加椒盐噪声');
subplot(2,2,4),imshow(J3),title('加乘性噪声');
```

例 4-2-8 的结果如图 4-2-9 所示。

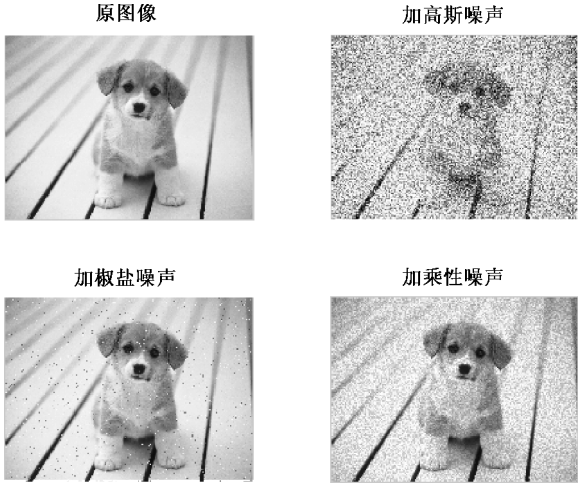


图 4-2-9　不同噪声的叠加

【例 4-2-9】维纳滤波。

程序如下：

```
I=imread('concordorthophoto.png');
figure (1);imshow(I);
J=imnoise(I,'gaussian',0,0.005); %施加高斯噪声
figure(2);imshow(J);
%Wiener 滤波
K=wiener2(J,[5 5]);
figure(3);imshow(K);
```

运行程序，通过观察可看出加入噪声的图像和维纳滤波后的图像与原图的区别。

4.2.6　图像变换

由于图像阵列很大，将空间域的处理转换为变换域处理，不仅可减少计算量，还可以给后续工作带来极大的方便，图像变换常用于图像压缩、滤波、编码和后续的特征抽取或信息分析过程。例如，傅里叶变换可使处理分析在频域中进行，使运算简单；而离散余弦变换可使能量集中在少数数据上，从而实现数据压缩，便于图像传输和存储。

傅里叶变换在图像处理中是一个最基本的数学工具，将时间信号正变换到频率域后进行处理（如低通、高通或带通），然后再反变换成时间信号，即可对图像的频谱进行各种各样的处理，它能够定量地分析诸如数字化系统、采样点、电子放大器、卷积滤波器、噪声和显示点等的作用，有助于解决大多数图像处理问题，如滤波、降噪、增强等。

实际中一般采用快速傅里叶变换（FFT）的方法，MATLAB 中的 fft2 指令用于得到二维 FFT 的结果，ifft2 指令用于得到二维 FFT 逆变换的结果。

在图像处理工具箱中，dct2 和 idct2 函数实现二维离散余弦变换及逆变换。大多数情况下，离散余弦变换（DCT）用于压缩图像，JPEG 图像格式就采用了 DCT 算法。在 JPEG 图像压缩算法中，图像被分成 8×8 或者 16×16 的图像块，然后对每个图像块进行 DCT 变

换。DCT 变换被量化、编码及传输。在接收端，量化的 DCT 系数被解码，并用来计算每个图像块的逆 DCT 变换，最后把各图像块拼接起来构成一幅图像。对一幅典型的图像而言，许多 DCT 变换的系数近似为 0，把它们去掉并不会明显影响重构图像的质量。

小波变换通过多分辨分析过程将一幅图像分成近似和细节两部分，细节对应的是小尺度的顺变，它在本尺度内很稳定，因此将细节存储起来，近似部分在下一个尺度进行分解，重复该过程即可，近似与细节在正交镜像滤波器算法中分别对应于高通滤波和低通滤波，这种变换通过尺度去掉相关性。因此，对信号的小波分解可以等效于信号通过了一个滤波器组，其中一个滤波器为低通滤波器，另一个为高通滤波器。MATLAB 工具箱中的 dwt 和 idwt 函数可实现一维离散小波变换及其反变换，wavedec 和 waverec 用于一维信号的多层小波分解和多层重构等。

【例 4-2-10】图像的傅里叶变换。

程序如下：

```
load imdemos saturn2
subplot(1,2,1);imshow(saturn2)
B=fftshift(fft2(saturn2));
subplot(1,2,2);imshow(log(abs(B)),[])
colormap(jet(64)), colorbar
```

例 4-2-10 的结果如图 4-2-10 所示。

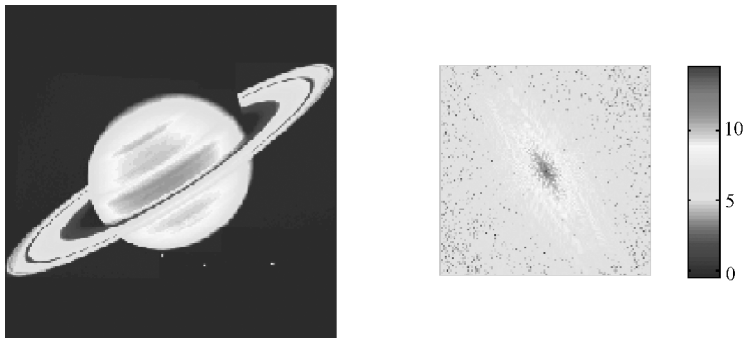


图 4-2-10　图像的傅里叶变换

【例 4-2-11】图像的 DCT 变换。

程序如下：

```
a=imread('onion.png');
i=rgb2gray(a);
j=dct2(i);
subplot(131);imshow(log(abs(j))),colorbar
j(abs(j)<10)=0;
k=idct2(j);
subplot(132);imshow(i);
subplot(133);imshow(k,[0,255]);
info=imfinfo('trees.tif')    %显示图像信息
```

例 4-2-11 的结果如图 4-2-11 所示。

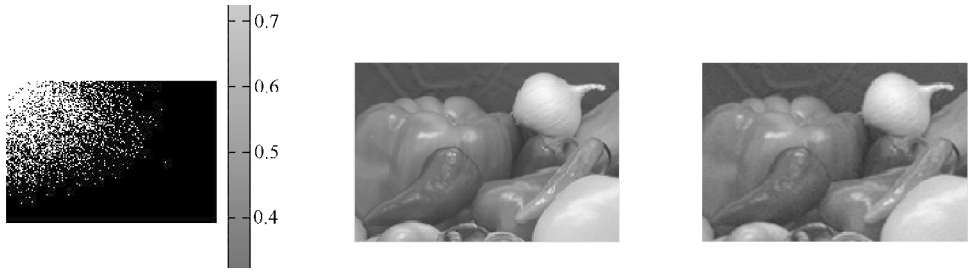


图 4-2-11　离散余弦变换

4.2.7　图像压缩

图像的数据量特别大，同时现在对图像需求的增长超过了网络带宽的限制，所以压缩是图像传输和存储的一个关键技术。图像压缩是指以较少的比特有损或无损地表示原来的像素矩阵的技术，也称为图像编码。图像数据之所以能被压缩，就是因为数据中存在着冗余。图像数据的冗余主要表现为：

- 图像中相邻像素间的相关性引起的空间冗余；
- 图像序列中不同帧之间存在相关性引起的时间冗余；
- 不同彩色平面或频谱带的相关性引起的频谱冗余。

图像压缩可以采用有损数据压缩也可以采用无损数据压缩。对于绘制的技术图、图表或者其他有价值的图像为避免失真优先使用无损压缩，要求不是很高的图像，如自然的图像可采用有损压缩，减少存储空间。在 MATLAB 中可调用多种方法进行压缩，如使用图像比例变换缩放图像、DCT 变换、小波变换等实现图像压缩。

在 JPEG 图像压缩法里，要将一幅彩色图像进行压缩编码，首先将图像 RGB 分量转化为亮度分量和色差分量，然后将图像分成 8×8 的像素块，用正向二维 DCT 把每个块转变成 64 个 DCT 系数值，其中 1 个数值是直流（DC）系数，即 8×8 空域图像子块的平均值，其余的 63 个是交流（AC）系数，接下来对 DCT 系数进行 Zig-Zag 扫描和 Huffman 编码，实现了图像压缩。

小波变换将图像的像素解相关的变换系数进行编码，比对元像素本身编码的效率更高。如果变换的基函数将大多数重要的可视信息压缩到少量的系数中，则剩下的系数可以被粗略的量化或截取为 0，而图像几乎没有失真。它的压缩比高，压缩速度快，压缩后能保持信号与图像的特征不变，且在传递中抗干扰。

【例 4-2-12】DCT 变换用于图像压缩。

程序如下：

```
I=imread('cameraman.tif');
I=im2double(I);     %图像存储类型转换，原图为灰度图像
T=dctmtx(8);        %离散余弦变换矩阵
B=blkproc(I,[8 8],'P1*x*P2',T,T');
%对原图像进行 DCT 变换
```

```
mask=[1 1 1 1 0 0 0 0
1 1 1 0 0 0 0 0
1 1 0 0 0 0 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0];
B2=blkproc(B,[8 8],'P1.*x',mask);

I2=blkproc(B2,[8 8],'P1*x*P2',T',T);

subplot(1,2,1);imshow(I)
title('原始图像')
subplot(1,2,2);imshow(I2)
title('压缩后的图像')
```

例 4-2-12 的结果如图 4-2-12 所示。

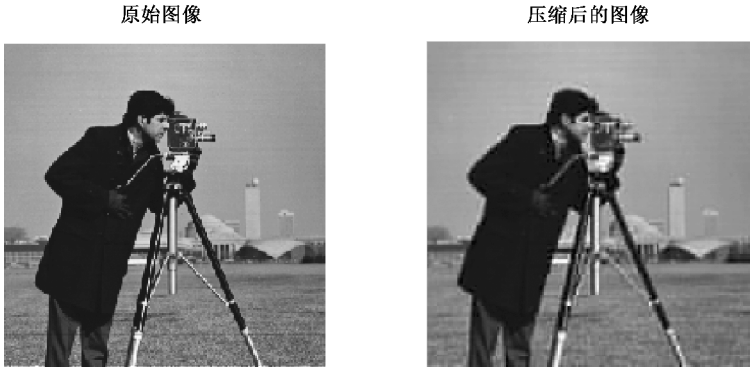


图 4-2-12　DCT 变换实现图像压缩

【例 4-2-13】小波变换实现图像压缩。

程序如下：

```
clear;
load wbarb;
subplot(3,3,1);image(X);colormap(map);
title('原始图像');
disp('原始图像 X 的大小：');
whos('X');
[c,s]=wavedec2(X,2,'bior3.7');
cal=appcoef2(c,s,'bior3.7',1);
ch1=detcoef2('h',c,s,1);
cv1=detcoef2('v',c,s,1);
cd1=detcoef2('d',c,s,1);
a1=wrcoef2('a',c,s,'bior3.7',1);
h1=wrcoef2('h',c,s,'bior3.7',1);
```

```
v1=wrcoef2('v',c,s,'bior3.7',1);
d1=wrcoef2('d',c,s,'bior3.7',1);
c1=[a1,h1;v1,d1];
ca1=appcoef2(c,s,'bior3.7',1);
ca1=wcodemat(cal,440,'mat',0);
ca1=0.8*cal;
subplot(3,3,2);image(ca1);colormap(map);
axis square;
title('第一次压缩图像 0.8 倍');
disp('第一次压缩图像的大小');
whos('ca1');
ca2=appcoef2(c,s,'bior3.7',2);
ca2=wcodemat(ca2,440,'mat',0);
ca2=0.6*ca2;
subplot(3,3,3);image(ca2);colormap(map);
axis square;
title('第二次压缩图像 0.6 倍');
disp('第二次压缩图像的大小');
whos('ca2');
ca3=appcoef2(c,s,'bior3.7',2);
ca3=wcodemat(ca3,440,'mat',0);
ca3=0.4*ca3;
subplot(3,3,4);image(ca3);colormap(map);
axis square;
title('第三次压缩图像 0.4 倍');
disp('第三次压缩图像的大小');
whos('ca3');ca3=appcoef2(c,s,'bior3.7',2);
ca3=wcodemat(ca3,440,'mat',0);
ca4=appcoef2(c,s,'bior3.7',2);
ca4=wcodemat(ca4,440,'mat',0);
ca4=0.2*ca4;
subplot(3,3,5);image(ca4);colormap(map);
axis square;
title('第四次压缩图像 0.2 倍');
disp('第四次压缩图像的大小');
whos('ca4');ca4=appcoef2(c,s,'bior3.7',2);
ca4=wcodemat(ca4,440,'mat',0);
ca5=appcoef2(c,s,'bior3.7',2);
ca5=wcodemat(ca5,440,'mat',0);
ca5=0.09*ca5;
subplot(3,3,6);image(ca5);colormap(map);
axis square;
title('第五次压缩图像 0.09 倍');
disp('第五次压缩图像的大小');
whos('ca5');ca5=appcoef2(c,s,'bior3.7',2);
ca5=wcodemat(ca5,440,'mat',0);
ca6=appcoef2(c,s,'bior3.7',2);
```

```
ca6=wcodemat(ca6,440,'mat',0);
ca6=0.04*ca6;
subplot(3,3,7);image(ca6);colormap(map);
axis square;
title('第六次压缩图像 0.04 倍');
disp('第六次压缩图像的大小');
whos('ca6');ca6=appcoef2(c,s,'bior3.7',2);
ca6=wcodemat(ca6,440,'mat',0);
ca7=appcoef2(c,s,'bior3.7',2);
ca7=wcodemat(ca7,440,'mat',0);
ca7=0.02*ca7;
subplot(3,3,8);image(ca7);colormap(map);
axis square;
title('第七次压缩图像 0.02 倍');
disp('第七次压缩图像的大小');
whos('ca7');ca2=appcoef2(c,s,'bior3.7',2);
ca7=wcodemat(ca2,440,'mat',0);
ca8=appcoef2(c,s,'bior3.7',2);
ca8=wcodemat(ca8,440,'mat',0);
ca8=0.01*ca8;
subplot(3,3,9);image(ca8);colormap(map);
axis square;
title('第八次压缩图像 0.01 倍');
disp('第八次压缩图像的大小');
whos('ca8');ca8=appcoef2(c,s,'bior3.7',2);
ca8=wcodemat(ca8,440,'mat',0);
```

例 4-2-13 的结果如图 4-2-13 所示。

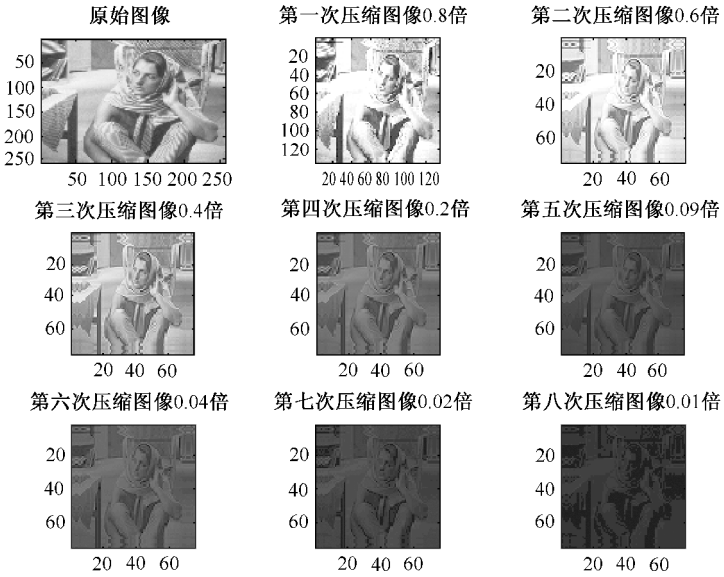


图 4-2-13　小波变换实现图像压缩

【例 4-2-14】小波变换压缩及压缩参数显示。

程序如下：

```
load wbarb;
subplot(2,2,1);
image(X);colormap(map);
[c,l]=wavedec2(X,2,'db3');
[thr,sorh,keepapp]=ddencmp('cmp','wv',X);
[Xcmp,cxc,lxc,perf0,perf12]=wdencmp('gbl',c,l,'db3',2,thr,sorh,keepapp);
subplot(2,2,2);
image(Xcmp);colormap(map);
disp('小波分解系数中为 0 的系数个数百分比：');
perf0
disp('压缩后保留能量百分比：');
perf12
```

例 4-2-14 的结果如图 4-2-14 所示。

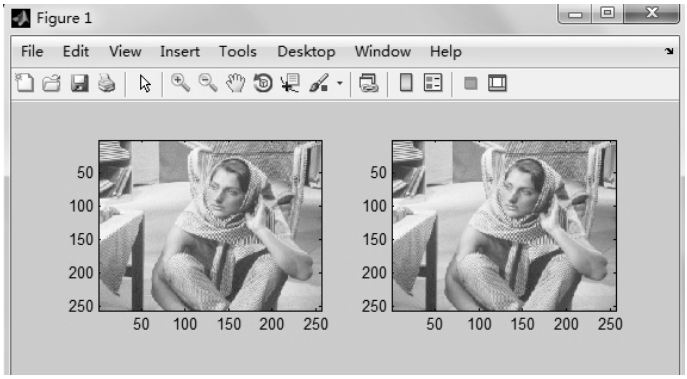


图 4-2-14　wdencmp 函数实现图像压缩

4.2.8　图像分割

图像分割是将图像中有意义的特征部分提取出来，其有意义的特征有图像中的边缘、区域等，这是进一步进行图像识别、分析和理解的基础。分割的方式有：

（1）按幅度不同来分割各个区域，即幅度分割；

（2）按边缘不同来划分各个区域，即边缘检测；

（3）按形状不同来分割各个区域，即区域分割。

边缘是图像的一个基本特征，携带了图像中的大量信息，边缘检测不仅能得到关于边界的有用的结构信息，而且还能极大地减少要处理的数据，很多图像处理和识别算法都以边缘检测为重要基础。

边缘检测是利用物体和背景在某种图像特性上的差异来实现的，这些差异包括灰度、颜色或者纹理特征，它是一种重要的区域处理方法。边缘是所要提取目标和背景的分界线，提取出边缘才能将目标和背景区分开来。

边缘检测包括两个基本内容：一是抽取出反映灰度变化的边缘点；二是剔除某些边界点或填补边界间断点，并将这些边缘连接成完整的线。如果一个像素落在边界上，那么它的邻域将成为一个灰度级变化地带。对这种变化最有用的两个特征是灰度的变化率和方向。边缘检测算子可以检查每个像素的邻域，并对灰度变化率进行量化，也包括对方向的确定，其中大多数是基于方向导数掩模求卷积的方法。MATLAB 工具箱提供的 edge 函数可针对 sobel 算子、prewitt 算子、roberts 算子、log 算子和 canny 算子实现检测边缘的功能。

【例 4-2-15】图像的边缘化处理。

程序如下：

```
I=imread('coins.png');
BW1=edge(I,'roberts');
BW2=edge(I,'sobel');
BW3=edge(I,'log');
figure
subplot(221),imshow(I),title('原图')
subplot(222),imshow(BW1),title('roberts 算子')
subplot(223),imshow(BW2),title('sobel 算子')
subplot(224),imshow(BW3),title('laplacian 算子')
```

例 4-2-15 的结果如图 4-2-15 所示。

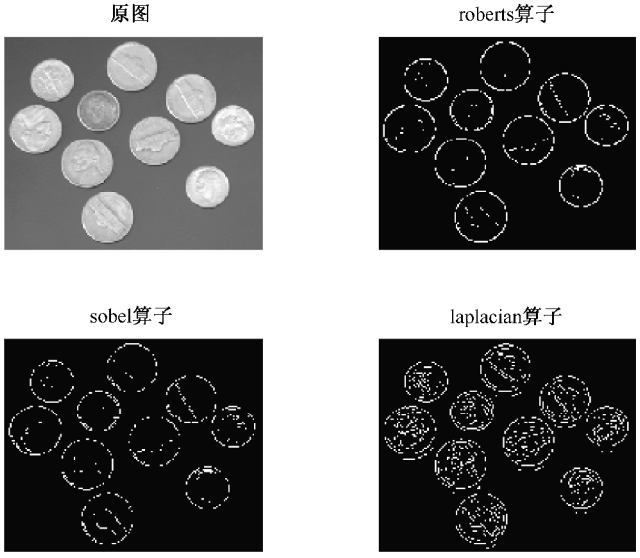


图 4-2-15　图像的边缘化处理

4.2.9　图像识别

图像识别，是指利用计算机对图像进行处理、分析和理解，以识别各种不同模式的目标和对象的技术，属于人工智能领域范畴。其主要内容是图像经过某些预处理（增强、复原、压缩）后，进行图像分割和特征提取，从而进行判决分类。图像识别中用到的方法很多。

【例 4-2-16】汽车牌照定位与字符识别。待处理的图像如图 4-2-16 所示，要求定位车牌位置并识别车牌字符。

图 4-2-16　汽车原图

该图像整体比较清晰干净，车牌方向端正，字体清楚，与周围颜色的反差较大。处理思路如下：

（1）首先选择恰当的门限值，转换为二值图像，提取出显示车牌的白色部分。

（2）根据二值图像的显示特点，去掉车身上部分肯定不是车牌位置的区域。

（3）将车牌图像反白处理，提取出车牌中的字符。

（4）从文件读取字符模板。对图像计算傅里叶描述子，用预先定义好的决策函数对描述子进行计算。进行模板匹配的相关运算，即计算两个图像矩阵之间的相关系数。变换后的图像中，亮度的高低指示相应区域与模板的匹配程度。

程序如下：

```
I=imread('car.jpg');
I2=rgb2gray(I);
I4=im2bw(I2, 0.2);                %转化为二值图像，采用门限值为 0.2
bw=bwareaopen(I4, 500);           %去除图像中面积过小肯定不是车牌的区域。
se=strel('disk',15);              %将白色区域膨胀，腐蚀去无关的小物件，包括车牌字符
bw=imclose(bw,se);
bw=imfill(bw,[1 1]);
[B,L]=bwboundaries(bw,4);  %查找连通域边界。保留此图形，备做标记
imshow(label2rgb(L, @jet, [.5 .5 .5]))
hold on
for k=1:length(B)
    boundary=B{k};
    plot(boundary(:,2),boundary(:,1),'w','LineWidth',2)
end
%找到每个连通域的质心
stats=regionprops(L,'Area','Centroid');
%循环历遍每个连通域的边界，依据车牌的尺寸找出连通域中最可能是车牌的一个
for k=1:length(B)
    %获取一条边界上的所有点
    boundary=B{k};
```

```
        %计算边界周长
        delta_sq=diff(boundary).^2;
        perimeter=sum(sqrt(sum(delta_sq,2)));
        %获取边界所围面积
        area=stats(k).Area;
        %计算匹配度
        metric=27*area/perimeter^2;
        %要显示的匹配度字串
        metric_string=sprintf('%2.2f',metric);
        %标记出匹配度接近 1 的连通域
        if metric>=0.9 && metric <= 1.1
            centroid=stats(k).Centroid;
            plot(centroid(1),centroid(2),'ko');
            % 提取该连通域所对应在二值图像中的矩形区域
            goalboundary=boundary;
            s=min(goalboundary, [], 1);
            e=max(goalboundary, [], 1);
            goal=imcrop(I4,[s(2) s(1) e(2)-s(2) e(1)-s(1)]);
        end
        %显示匹配度字串
        text(boundary(1,2)-35,boundary(1,1)+13,...
        metric_string,'Color','g',...
        'FontSize',14,'FontWeight','bold');
    End
    goal=~goal;                    %将车牌图像反白处理，并扩充为 256×256 的方阵
    goal(256,256)=0;
    figure;
    imshow(goal);
    w=imread('P.bmp');             %从文件读取字符模板，对图像计算傅里叶描述子
    w=~w;
    C=real(ifft2(fft2(goal).*fft2(rot90(w,2),256,256)));
    thresh=240;                    %通过检查 C 的最大值，试验确定一个合适的门限
    figure;
    imshow(C>thresh);
```

例 4-2-16 的结果如图 4-2-17、图 4-2-18 和图 4-2-19 所示。

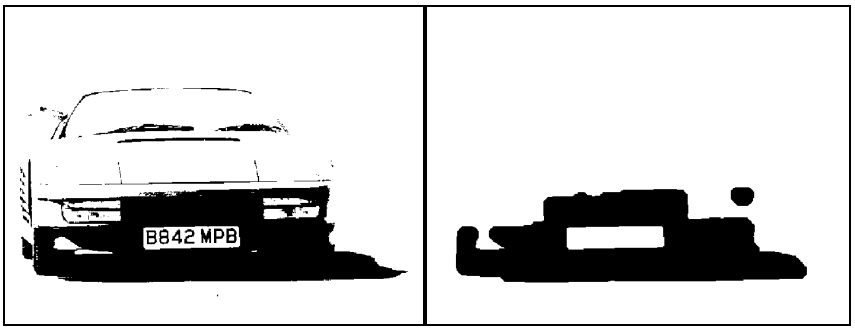


图 4-2-17　第一次处理

图 4-2-18　第二次处理

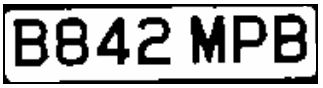


图 4-2-19　第三次处理

程序中以字符 P 为例对照模板进行定位识别，同样的方法，可以识别和定位其他字符。

第 5 章　系统建模及仿真算法

仿真的第一步就是建立与实际系统相符合的仿真模型。如何抽象提取系统中用户有用的因素，用已知的知识体系描述并求解是建模中的关键环节。模型建立后如何在计算机上通过仿真工具实现，则依赖于工具的计算能力和特点。MATLAB 可通过编程或模块化方式建立系统的静态和动态模型，数值积分法是 MATLAB 中仿真算法的核心。

5.1　系统和模型

5.1.1　系统

1．系统的定义

系统是由若干要素以一定结构形式连接构成的具有某种功能的有机整体，是一个内涵十分丰富的概念，泛指自然界的一切现象与过程，大至无限的宇宙世界，小到分子、原子，都可称之为系统。

任何系统都可分解为以下部分。

- 实体：组成系统的元素、对象。实体确定了系统的构成，也确定了系统的边界；
- 属性：实体的特征；
- 活动：系统由一个状态到另一个状态的变化过程。活动定义了系统内部实体之间的相互作用，反映了系统内部发生变化的过程；
- 状态：在任意时刻，系统中实体、属性、活动的信息总和。

仿真中对系统的研究一般都需考虑如下特征。

- 组成性：系统由两个或两个以上要素组成；
- 层次性：系统要素应该能够区分；
- 边界性：要素的边界小于系统的边界；
- 相关性：要素相互联系，要素和系统都是相对的；
- 目的性：要素的结合是为了达到特定的目的；
- 整体性：系统是一个整体。

研究系统主要研究系统的动态变化，除了研究系统的实体属性活动外，还需要研究影响系统活动的外部条件，这些外部条件称之为环境。但系统与环境的边界是不确定的，对于相同系统可能因为不同的研究目的而有不同的环境。

2．系统的分类

实际应用中，可根据不同的需要将系统分成各种类型。例如：

- 工程系统、非工程系统；
- 连续系统、离散事件系统；
- 白色系统、灰色系统、黑色系统；
- 简单系统、复杂系统；
- 小系统、大系统、巨系统。

不同的学科也可根据其特点将系统进一步细化分类。例如，在控制系统仿真中对系统通常有以下几种分类方法。

1）按物理特征

- 工程系统：如电气控制系统、机械控制系统、化工控制系统、热力控制系统、冶金控制系统等。
- 非工程系统：如交通管理系统、生态控制系统、大气监测系统等。

2）按复杂程度

- 简单系统：系统中只有一个外部输入信号，输出也为一个信号，通常有一个主反馈通路，内部可以带有多个局部反馈，也称为单输入单输出系统。
- 复杂系统：系统信号多、回路多，相互之间有耦合作用，是一种多变量系统，也称为多输入多输出系统，通常采用计算机控制。例如，飞行器的状态控制，冶炼厂的化学反应过程控制，大气及环境变化的监测控制，生态平衡监测控制等。

3）按数学模型

- 线性系统：系统中各元件的特性为直线，可以采用叠加原理，其数学模型表现为线性微分方程式。
- 非线性系统：系统中包含有非线性特性，例如，饱和、继电、摩擦、滞环等，系统特性与线性系统存在着本质的差别。通常采用非线性的描述函数表示。

4）按信号的变化规律

- 线性连续系统：系统的状态随时间线性连续地变化，其动态特性可采用微分方程或状态方程来描述。
- 采样系统：系统中存在脉冲序列或数码信号，其数学模型采用差分方程来描述，也称为离散系统。
- 离散事件系统：系统中的状态变化只在离散时刻发生，而且往往是随机性的，通常采用“事件”来表征其变化，采用流程图或网络图来表征系统模型，如服务台排队系统、飞机订票系统、库存管理系统、交通控制系统等。

5）按输出特征

- 随动系统：系统的输出量按照一定的精度跟随参考输入量的变化，也称为伺服系统或跟踪系统，如雷达天线随动系统、火炮发射跟踪控制系统、自动测量仪器系统、电力拖动控制机械装置系统等。

- 自动稳定系统：系统在任何内、外部扰动作用下，其输出值保持为恒定，也称为恒值控制系统，如反馈式稳压、稳流装置，恒转速控制系统、恒温度控制系统等。
- 程序控制系统：系统的输入为时间函数，输出跟随输入按程序规定的要求变化，如机床数控系统、绘图仪控制系统等。
- 数字控制系统：系统的模拟输入量经 A/D 转换成数字量，在控制器的作用下，再将数字量经 D/A 转换变成模拟量输出，如常见的计算机控制系统。

5.1.2 模型

1．模型的定义

仿真是研究系统的一种重要手段，而系统模型则是仿真所要研究的直接对象。

模型是系统的物理的、数学的或其他方式的逻辑表述，它以某种确定的形式（符号、文字、图表、实物、数学公式）提供关于系统的知识。

从某种意义上说，模型是系统的代表，同时也是对系统的简化。由于不同的分析者所关心的是系统的不同方面，或者由于同一分析者要了解系统的各种变化关系，对同一个系统可以产生相应于不同层次的多种模型。

2．模型的分类

一般来说，模型可分为以下三类。

（1）概念模型：对系统的一种定性描述，用于表示系统组成和相互关系，如功能模型、结构模型等。

（2）物理模型：实物模型，如航模、船模。

（3）数学模型：采用数学语言对系统进行定量描述。

计算机仿真中所针对的模型主要是指数学模型。

3．建模的原则

建模中应把握以下原则。

（1）抓住主要矛盾。建模都是针对某一目的而言的，所以建模时只应包括与研究目的有关的方面，抓住问题的主要方面，而不是涵盖对象系统的所有方面。例如，对于一个空运指挥调度系统的研究，建模时只需考虑飞机的飞行航向而无须考虑其飞行姿态。

（2）力争清晰明了。在现实生活中，需要研究的对象往往是非常复杂的大系统，如社会经济系统、环境系统等。一个大型复杂系统是由许多联系密切的子系统组成的，而且子系统有时也包含自己的子系统，层层叠加，使得结构非常复杂，给研究带来了巨大的困难。这就要求在建模时，子模型与子模型之间，除了保留研究目的所必要的信息联系外，其他的耦合关系要尽可能减少，以保证模型结构尽可能清晰明了。

（3）精度要求适当。精度要求是模型的一个重要方面，其要求的高低对系统模型有重要的影响。但是，并非精度越高越好。建立系统模型时，应该视研究目的和使用环境不同，选择适当的精度等级，以保证模型切题、实用，而又不致于花费太多。

（4）尽量使用标准模型。建立一个实际系统的模型时，应该首先大量调阅模型库中的

标准模型，使用标准模型，或者尽可能向标准模型靠拢。这样有利于比较分析，可节省费用和时间。

4．建模的方法

建模通常有如下几种方法。

（1）分析法（演绎法/理论建模/机理模型）。分析法是根据系统的工作原理，运用一些已知的定理、定律、原理推导出系统的数学模型。分析法主要针对内部结构和特性已经清楚的系统，即所谓的“白箱”系统，使用该方法的前提是对系统的运行机理完全清楚。

（2）测试法（归纳法/试验建模/系统辨识 ）。测试法是通过实验方法，选择各种典型的输入量，并测试其相应的输出量，然后按照一定的辨识方法，得到系统模型。测试法主要针对内部结构和特性不清楚或者难以弄清楚的系统，即所谓的“黑箱”或“灰箱”系统，测试法建模通常只用于建立输入/输出模型。测试建模法又可分为经典辨识法和系统辨识法两大类：

① 经典辨识法。经典辨识法不考虑测试数据中偶然性误差的影响，只需对少量的测试数据进行比较简单的数学处理，其计算工作量一般较小。经典辨识法包括时域法、频域法和相关分析法。

② 系统辨识法。系统辨识就是在输入和输出数据的基础上，从一组给定的模型类中，确定一个与所测系统等价的模型。其特点是可以清除测试数据中的偶然性误差即噪声的影响，为此需要处理大量的测试数据，计算机是不可缺少的工具。

（3）统计分析法。对于那些属于“黑箱”，但又不允许直接进行实验观察的系统，可以采用数据收集和统计分析的方法来建造系统模型。统计法使用的前提是必须有足够正确的数据，所建的模型也只能保证在这个范围内有效；足够的数据不仅仅指数据量多，而且数据的内容要丰富（频带要宽），能够充分激励要建模系统的特性；

大部分系统模型的构建往往是上述几种方法综合运用的结果，如一个复杂系统有多个子系统，则多个子系统可能就会用不同的方法进行分析，最终整合成一个总体模型。

在实际应用中，要清楚每种方法的局限性，掌握适用范围，进行组合采用、互补。

5.1.3 数学模型

1．数学模型分类

用抽象的数学语言描述系统内部物理变量之间的关系而建立起来的模型，称为该系统的数学模型。通过对系统的数学模型的研究可以揭示系统的内在运动和系统的动态性能。数学建模就是确定系统的模型形式、结构和参数，以得到正确描述系统表征和性状的最简数学表达式。数学模型又可以分为静态模型和动态模型两类。

1）静态模型

静态模型与时间没有关系，模型的一般形式是代数方程、逻辑表达关系式。如理想电位器的转角和输出电压之间的关系式或继电器的逻辑关系式等。

2）动态模型

在动态模型中，时间扮演着不可或缺的角色。实际系统仿真所模拟的对象多数是动态系统。动态模型又可分为连续系统动态模型和离散系统动态模型。

（1）连续系统动态模型。在此模型中，系统状态随时间连续变化，如水库蓄水量、放水量以及出现降水和蒸发时水位的变化均属此类，连续系统动态模型有确定型模型和随机型模型两种。

① 确定型系统。若一个系统的输出完全可以用它的输入来表示，则称为确定性系统。确定型模型又分为集中参数模型和分布参数模型两种。集中参数模型描述系统运动用的是常微分方程、状态方程和传递函数；而描述热传递过程的偏微分方程则是典型的分布参数模型。

② 随机型模型。若系统的输出是随机的，即对于给定的输入存在多种可能的输出，则该系统是随机系统。

（2）离散系统动态模型。在此模型中，系统状态仅在离散的时刻点发生变化，例如在制造系统中，零件会在特定的时间到达和离开，机器会在特定的时刻进行运动操作。

离散模型又可进一步分为时间离散系统模型和离散事件模型。

① 时间离散系统模型：时间离散系统又称为采样控制系统，一般用差分方程、离散状态方程和脉冲传递函数来描述。这种系统的特性其实是连续的，仅仅在采样的时刻点上来研究系统的输出，如各种数字式控制器的模型均居于这一类。

② 离散事件模型：离散事件系统的输出不完全由输入作用的形式描述，往往存在着多种可能的输出。它是一个随机系统，如库存系统、管理车辆流通的交通系统、排队服务系统等。输入和输出在系统中是随机发生的，一般要用概率模型来描述这种系统。

在有的模型中，既有连续变化的成分，也有离散变化的因素，这种模型被称为混合模型，例如在炼油厂，储油罐中的压力是连续变化的，但会在离散时间点上发生间歇。

数学模型与表达形式见表 5-1-1，模型的分类如图 5-1-1 所示。

表 5-1-1　数学模型与表达形式

系 统 特 性	数 学 描 述	系 统 特 性	数 学 描 述
线性	线性方程	集中参数	常微分方程
非线性	非线性方程	分布参数	偏微分方程
静态	代数方程	连续	微分方程
动态方程	微分、差分方程	离散	差分方程
确定性	不含随机变量	参数	数学表达式
随机性	含随机变量	非参数	图、表
微观	微分、差分方程	时域	微分、差分方程
宏观	代数、积分方程	频域	频率特性
定常	不含对时间的系数	输入输出	传递函数微分方程
非定常	含对时间的系数	状态空间	状态方程

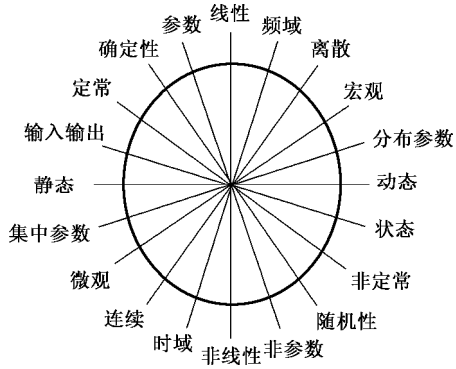


图 5-1-1　数学模型的分类

一般来说，系统数学模型的建立需要建立如下抽象：输入变量、输出变量、状态变量及相互间的函数关系，这种抽象过程称为模型构造。抽象中，必须联系真实系统与建模目标，其中描述变量起着很重要的作用，它可观测或不可观测。

输入变量是从外部对系统施加影响或干扰的可观测变量，输出变量是系统对输入变量的响应结果。输入、输出变量对的集合表征着真实系统的“输入-输出”性状（关系）。

能够完全描述动态系统时域行为的所含变量个数最少的变量组称为系统的状态变量。所谓完全描述系统的时域行为指的是，如果给定初始时刻 $t_0 \in I$ 的状态 $x(t_0)$和$[t_0, t] \in I$ 上的输入函数 $u(t)$，则系统在$[t_0, t]$上任何瞬时的行为被唯一确定。

2．连续系统的数学模型

连续系统广泛存在于航空、航天、动力、控制、化工等领域中，虽然其数学模型很多，但大体可归为三类，即连续时间模型、离散时间模型及连续-离散混合模型。

1）连续时间模型

假定一个系统的输入量 $u(t)$、输出量 $y(t)$及状态内部状态变量 $x(t)$都是时间 t 的连续函数，常用以下四种数学模型形式描述。

（1）微分方程：

$$\frac{\mathrm{d}^t y(t)}{\mathrm{d}t^n} + a_1 \frac{\mathrm{d}^{n-1} y(t)}{\mathrm{d}t_{n-1}} + \cdots + a_n y(t) = c_0 \frac{\mathrm{d}^{n-1} u(t)}{\mathrm{d}t^{n-1}} + c_1 \frac{\mathrm{d}^{n-2} u(t)}{\mathrm{d}t^{n-2}} + \cdots + c_n u(t)$$

即

$$(p^{(n)} + a_1 p^{(n-1)} + \cdots + a_n) y(t) = (c_0 p^{(n-1)} + c_1 p^{(n-2)} + \cdots + c_{n-1}) u(t)$$

式中，$p \overset{\text{dif}}{=} \dfrac{d}{\mathrm{d}t}$ 为微分算子。

（2）传递函数：

$$G(s) = \frac{Y(s)}{U(s)} = \frac{c_0 s^{n-1} + c_1 s^{n-2} + \cdots + c_{n-1}}{s^n + a_1 s^{n-1} + \cdots + a_n}$$

式中，s 为拉普拉斯算子。

（3）权函数：在零初始条件下，系统对理论脉冲函数 $\delta(t)$输入的响应 $g(t)$称为权函数，又称为脉冲过渡函数。这里，$\delta(t)=\begin{cases}\infty, & t=0\\ 0, & t\neq 0\end{cases}$，且 $\int_0^{\infty}\delta(t)\mathrm{d}t=1$。

可以证明：$g(t)=L^{-1}[G(s)]$，且对于任意函数 $u(t)$作用的系统响应 $y(t)$，可用卷积描述，即

$$y(t)=\int_0^{\tau}u(\tau)g(t-\tau)\mathrm{d}t$$

以上三种模型仅描述系统的输入量与输出量之间的关系，并未涉及内部情况，故称为系统的外部模型。

（4）状态空间表达式：由状态方程和输出方程构成，其矩阵形式为

$$\begin{cases}\dot{x}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}u\\ y=\boldsymbol{C}\boldsymbol{x}\end{cases}$$

式中，$\boldsymbol{A}$ 为状态系数阵，$\boldsymbol{B}$ 为输入系数阵，$\boldsymbol{C}$ 为输出系数阵，$\boldsymbol{x}$ 为状态阵。状态空间表达式是系统的内部模型。

$$\boldsymbol{A}=\begin{bmatrix}0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & 1\\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1\end{bmatrix}$$

$$\boldsymbol{B}=\begin{bmatrix}0\\ 0\\ \vdots\\ 0\\ 1\end{bmatrix}$$

$$\boldsymbol{C}=[c_{n-1}\quad c_{n-2}\quad \cdots\quad c_0]$$

$$\boldsymbol{x}=\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}$$

2）离散时间模型

假定一个系统的输入量、输出量和状态变量是时间的离散函数，即分别为时间序列 $\{u(k)\}$，$\{y(k)\}$，$\{x(k)\}$，那也亦有以下四种模型形式。

（1）差分方程：

$$a_n y(n+k)+a_k y(n+k-1)+\cdots+a_n y(k)=b_1 u(n+k-1)+\cdots+b_n u(k)$$

（2）z 传递函数（脉冲传递函数）：

$$H(z)=\frac{Y(z)}{U(z)}=\frac{b_1z^{-1}+\cdots+b_nz^{-n}}{a_0+a_1z^{-1}+\cdots+a_nz^{-n}}$$

（3）权序列：

$$y(k)=\sum_{i=0}^{k}u(i)h(k-i)$$

可以证明，$Z[h(k)]=H(z)$。

（4）离散状态空间表达式：

$$x(k+1)=\boldsymbol{F}X(k)+\boldsymbol{G}u(k)$$

$$y(k)=\sum_{j=1}^{n}b_jq^jx(k+n)=\sum_{j=1}^{n}b_jx_{n-j+1}(k)=\boldsymbol{\Gamma}x(k)$$

式中，

$$\boldsymbol{F}=\begin{bmatrix}0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \cdots & \cdots & \cdots & \ddots & \cdots\\ 0 & 0 & 0 & \cdots & 1\\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1\end{bmatrix}$$

$$\boldsymbol{G}=\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}$$

$$\boldsymbol{\Gamma}=[b_nb_{n-1}\cdots b_1]$$

3）连续-离散混合模型

如果系统的环节有的状态变量是连续变量，而有的是离散变量，即系统由离散和连续两部分组成，那么该系统使用连续及离散时间两类模型来描述。例如，现代计算机控制系统就是典型的连续-离散混合模型

3．非线性系统的数学模型

所有系统都具有一定程度的非线性，非线性可分固有（自然）非线性和外加（人为）非线性。若一个系统的工作范围较小，且包含的非线性较光滑，那么该系统可由某个线性化系统来适当逼近，而这个线性化系统的数学模型形式可用前述任何一种线性系统模型形式来描述。

具有严重非线性，不可线性化的系统称为本质非线性系统，对于这类系统，就必须引进非线性特性的数学描述，即建立系统的非线性模型。

非线性系统的基本数学模型是非线性方程，非线性方程可以是代数方程、微分方程、差分方程等。表 5-1-2 给出这些模型形式与非线性系统类型的一般关系。

表 5-1-2　非线性系统类型与方程形式的关系

方程形式 \ 系统类型		非线性系统类型
非线性方程形式	代数方程	具有静态非线性特性的系统
	常微分方程	集中参数的非线性系统
	偏微分方程	分布参数的非线性系统
	差分方程	非线性离散时间系统

另外，根据数学特性还可以把非线性分为连续非线性和断续非线性，由于断续非线性不能由线性函数局部逼近，因此又把它称为“强”非线性，主要有饱和特性、间隙特性、不灵敏区特性、继电特性、干摩擦或黏性摩擦特性、变增益特性、多变量非线性等。这些特性通常可用相应的描述函数来描述。

4．集中参数连续系统的数学建模

连续系统按照输入和输出是否与系统各质点位置有关可分为集中参数系统和分布参数系统。大多数系统严格讲属于分布参数系统，其输出不仅是时间的连续函数，而且也是系统各点位置的函数。因此，分布参数系统可视为由无限多个独立单元（或元件）构成，其特性决定于各个单元。根据建模目标的不同，往往许多分布参数系统可按照集中参数来处理。例如，一个物体是由无数个质点组成的，如果研究该物体的质心运动，便可假设成一个刚体，且认为每个质点的运动状态相同，整个物体的质量均集中在质心处，这样所得到的质心运动模型就是一个集中多数模型，类似集中参数问题建模在实际工程技术中是非常普遍的，如追踪问题，航天器发射与运行问题，汽车越野问题及高层建筑振动问题等，对于这类集中参数问题建模，一般是首先根据实际现象和大量试验数据作出适当假定，并构筑出它们的结构模态，然后在此基础上利用直接分析法建立起该系统的数学模型。

5．分布参数连续系统的数学建模

如果建模对象的每个微小部分的状态都不一样（如研究物体弹性振动、大气中污染物扩散等），那么这些系统就必须视为无限多个微小部分组成。这时，系统的输出将为多元函数，它不仅是时间的函数，而且是系统各点位置的函数，这种系统被称为分布参数系统。描述分布参数系统的数学模型通常是偏微分方程，该模型的建立一般要比上述集中参数连续系统复杂，多采用机理分析法和系统辨识方法。

5.2　仿真算法

5.2.1　算法的概念和性能

仿真算法是指在计算机上建立仿真模型进行仿真研究的方法，在系统模型转换为计算机模型中算法是核心问题。

在仿真中，针对具体系统选择算法需要关注算法的性能，对不同的算法进行比较与选择。例如，对刚性问题的求解就不能采用通常的积分算法。所谓刚性问题是指在用微分方

程描述的一个变化过程中，若往往又包含着多个相互作用但变化速度相差十分悬殊的子过程，这样一类过程就认为具有“刚性”，描述这类过程的微分方程初值问题称为“刚性问题”。例如，宇航飞行器自动控制系统一般包含两个相互作用但效应速度相差十分悬殊的子系统，一个是控制飞行器质心运动的系统，当飞行器速度较大时，质心运动惯性较大，因而相对来说变化缓慢；另一个是控制飞行器运动姿态的系统，由于惯性小，相对来说变化很快，因而整个系统就是一个刚性系统。

算法的性能分析包括误差、收敛性、计算效率等。例如微分方程常用的数值积分法有多种解法，每种解法的性能都有差异，要根据实际问题进行选择。

选择算法时首要考虑的是算法的稳定性。若运算过程中计算误差不断增长，算法为数值不稳定，反之则为稳定的。有关稳定性问题见 5.2.4 节。

此外，由于计算机是采用浮点数进行运算，误差在所难免。

采用浮点数运算存在的常见问题有

（1）舍入误差。计算机有一组操作浮点数的指令，用以模拟加、减、乘、除运算，但不可能精确。例如，乘法运算时，乘积应有 $2t$ 位精度，但实际仅能保留 t 位，即存在舍入误差。复杂计算（迭代等）中舍入误差的累积可能会影响结果，应在算法分析中考虑。

（2）溢出。如计算机对指数的取值范围限制，乘、除时的上溢、下溢，也应进行处理。

5.2.2 连续系统的仿真算法

连续系统的特点是系统的状态随时间连续变化，由于数字计算机的数值及时间均具有离散性，系统的数值及时间具有连续性，因此连续系统仿真的本质是从时间、数值两方面对原系统离散化，并选择合适的数值计算方法来近似积分运算，从而得到离散模型来近似原连续模型。

连续系统数字仿真的离散化方法有两类：

（1）数值积分方法，微分方程已知初值求解。

（2）离散相似方法，求连续系统的等价离散模型。

其中，数值积分方法是 MATLAB 中连续系统数字仿真中最基本的算法。

1．数值积分法

如前所述，连续系统可用微分方程数学模型来描述。高阶微分方程可转化为一阶微分方程组来研究，用一阶微分方程组初值问题解法来解高阶微分方程初值问题。因此，要对这类系统进行仿真，其实质就是采用数值积分法求解一阶微分方程。应用解析方法求解常微分方程初值问题，只能求出一些特殊类型的方程的解。在大多数情况下，只能通过其数值解满足工程需要。

此初值问题的数值解就是对未知函数 $y = f(x)$给出在一系列节点 $x_0, x_1, x_2, \ldots, x_n$ 处的函数值 $y_0, y_1, y_2, \ldots, y_n$。

用图形描述则是求出曲线下包含面积的近似值，如图 5-2-1 所示。

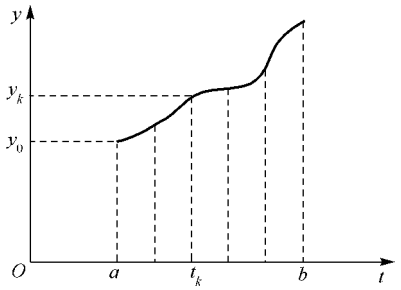


图 5-2-1　积分的图形描述

数值积分中常用的方法有：

（1）单步法和多步法。单步法指计算 y_{k+1} 值只需利用 t_k 时刻的信息，也称为自启动算法；多步法在计算 y_{k+1} 值时，则需利用 t_k，t_{k-1}，…时刻的信息。

（2）显示法和隐式法。显示法在计算 y_{k+1} 时所需数据均已算出；隐式法在计算 y_{k+1} 时需用到 t_{k+1} 时刻的数据，该算法必须借助预估公式。

（3）定步长和变步长。定步长为积分步长在仿真运行过程中始终不变；变步长指在仿真运行过程中自动修改步长。

对一阶常微分方程

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=f(x,y), & x\in[x_0,X]\\ y(x_0)=y_0\end{cases} \tag{5-2-1}$$

将方程 $y'=f(x,y(x))$ 两边对 x 从 x_i 到 x_{i+1} 积分，得

$$y(x_{i+1})=y(x_i)+\int_{x_i}^{x_{i+1}}f(x,y(x))\mathrm{d}x \tag{5-2-2}$$

用式（5-2-2）计算函数值 $y(x_{i+1})$ $y(x_{i+1})$ 的难点在于积分 $\int_{x_i}^{x_{i+1}}f(x,y(x))\mathrm{d}x$ 的计算，下面就用不同的数值积分公式代替此积分，得出相应的微分方程数值解公式, 常用的有欧拉法、梯形法、四阶龙格-库塔法等。

（1）欧拉法。欧拉（Euler）法是最简单的方法，但精度差，故不实用，然而对理论分析很有用。由于

$$\int_{x_i}^{x_{i+1}}f(x,y(x))\mathrm{d}x\approx hf(x_i,y(x_i))$$

式中，$h=x_{i+1}-x_i$ 为步长，于是有

$$y(x_{i+1})\approx y(x_i)+hf(x_i,y(x_i)) \tag{5-2-3}$$

用 y_i 替代上式右端的 $y(x_i)$，并将由式（5-2-3）右端算出的值 y_{i+1} 作为 $y(x_{i+1})$的近似值，这样建立的计算公式

$$y_{i+1}=y_i+hf(x_i,y_i),\qquad i=0,1,2,\cdots,n$$

称为欧拉公式，用此公式求解初值问题数值解的方法叫做欧拉方法。欧拉方法的几何意义是：在求积分的过程中，用矩形面积代替小区间的曲线积分。

（2）梯形法。为了提高精度，改用梯形公式计算求积项

$$\int_{x_i}^{x_{i+1}}f(x,y(x))\mathrm{d}x\approx\frac{h}{2}[f(x_i,y(x_i))+f(x_{i+1},y(x_{i+1}))] \tag{5-2-4}$$

称其为梯形公式，该公式的右端含有未知的 y_{i+1}，因此称其为隐式格式。其特点是：计算量小但不能自动开始，改变步长困难，为多步法。

为了应用式（5-2-4）计算 y_{i+1}，先用欧拉公式计算 y_{i+1} 的近似值 $\overline{y}_{i+1}$，称其为预报值，然后用预报值 $\overline{y}_{i+1}$ 代替式（5-2-4）中右端的 y_{i+1}，再计算出校正值 y_{i+1}。这样建立起来的预报-校正系统称为改进的欧拉公式，记为

$$\begin{cases} \overline{y}_{i+1} = y_i + hf(x_i, y_i) & \text{（预报）} \\ y_{i+1} = y_i + \dfrac{h}{2}[f(x_i, y_i) + f(x_{i+1}, \overline{y}_{i+1})] & \text{（校正），} \quad i = 0,1,2,\cdots,n \end{cases} \tag{5-2-5}$$

为了减少计算量，可将式（5-2-5）写为

$$\begin{cases} y_p = y_i + hf(x_i, y_i) \\ y_c = y_i + hf(x_{i+1}, y_p), \quad i = 0,1,2,\cdots,n \\ y_{i+1} = \dfrac{1}{2}(y_p + y_c) \end{cases}$$

（3）龙格-库塔法。Runge-Kutta 公式建立的基本思想是用不同点的函数值作线性组合构造近似公式。再将近似公式与解析解的 Taylor 展式相比较。要求二者前面的若干项吻合，从而使近似公式达到一定的阶数。

假定初值问题的解 $y(t)$及函数 $f(x, y(x))$充分光滑，则

$$y(x_{n+1}) = y(x_n) + y'y(x_n) + \frac{h^2}{2}y''y(x_n) + \cdots + \frac{h^p}{p!}y^{(p)}(x_n) + O(h^{p+1})$$

当 h 充分小时可略去余项 $O(h^{p+1})$ 时，则可导出 p 阶公式

$$\begin{cases} y_{n+1} = y_n + y'y_n + \dfrac{h^2}{2}y''y_n + \cdots + \dfrac{h^p}{p!}y_n^{(p)} \\ y_0 = y(x_0) \end{cases}$$

式中，各阶导数可表示为

$$u_n' = f(t_n, u_n)$$

$$u_n'' = f_t(t_n, u_n) + f(t_n, u_n)f_u(t_n, u_n)$$

$$u_n''' = \left[f_{tt} + 2f_{tu}f + f^2 f_{uu} + f_u^2 f + f_t f_u \right]|_{(t_n, u_n)}$$

该方法的缺点为计算高阶导数十分困难。

常用的二阶 R-K 方法有（Euler 预估校正格式）：

$$y_{n+1} = y_n + \frac{h}{2}(K_1 + K_2)$$

$$K_1 = f(x_n, yu_n)$$

$$K_2 = f(x_n + h, y_n + hK_1)$$

常用的四阶 R-K 方法有：

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + \dfrac{1}{2}h, y_n + \dfrac{1}{2}hk_1) \\ k_3 = f(x_n + \dfrac{1}{2}h, y_n + \dfrac{1}{2}hk_2) \\ k_4 = f(x_n + h, y_n + hk_3) \end{cases}$$

欧拉公式可以看做一阶龙格-库塔公式，二阶 Runge-Kutta 法计算精度高于 Euler 法，四阶 Runge-Kutta 法计算精度更高，经常采用。以上均为一步法。

Runge-Kutta 适合于高度非线性或不连续系统，不适合刚性系统。

微分方程式的刚性问题在性质上与代数方程式的病态问题相类似，它同动态系统中各环节时间常数大小范围的差别有很大关系。更确切地说，刚性问题是用线性化方程式的最大特征值和最小特征值的比率来衡量的。

刚性问题解答的难度就在于其快变子系统的干扰，当试图在慢变区间上求解刚性问题时，尽管快变分量的值已衰减到微不足道，但这种快速变化的干扰仍严重影响数值解的稳定性和精度，给整个计算带来很大的实质性的困难。

解决刚性系统的算法是 Gear 法，由于 Gear 法算法叙述繁琐，在此予以省略，读者有兴趣可参考相关书籍。

综合上述，数值积分常用算法的特点如下：

（1）Euler 方法是解初值问题最简单的数值方法，由于其精度低，一般不用于实际问题的求解。

（2）Runge-kutta 方法巧妙利用函数 $f(x, y)$在一些点上的函数值的线性组合，获得了高阶的数值解法，它避开了要获得高阶方法须对 $f(x, y)$求高阶导数的不便，是离散化方法中 Taylor 展开法的一个应用。Runge-kutta 方法主要用于定步长的情况，其中在准确性的工作量的综合效果看，经典的 Runge-kutta 方法是首选。Runge-kutta 方法也常用于对多步法提供初值。

（3）Adams 方法是线性多步法中常用的方法，具有计算函数 $f(x, y)$的次数少且精度高的特点，若计算 $f(x, y)$的代价高推荐用线性多步法。

（4）隐式方法一般比显式方法要好，由于其计算量较大，常用有这两种方法组合的预测-较正公式来近似代替隐式方法的使用。

【例 5-2-1】对下列传递函数分别采用 RK-4 和 Euler 法进行 MATLAB 仿真。

$$G_1(s)=\frac{1}{(5s+1)(10s+1)(20s+1)}$$

（1）利用 RK-4 法。

程序如下：

```
clear; clc;
num=[1];
den=conv(conv([5 1],[10 1]),[20 1]);
g1=tf(num,den);
[A,B,C,D]=ssdata(g1);
x0 = [0; 0; 0];
xk = x0;
u = 1;
h = 1/50;
ST = 100;
iST= round( ST/h );
```

```
t = zeros(iST,1);
y = zeros(iST,1);
for( ih = 1 : iST )
    t(ih+1) = t(ih) + h;
    K1 = A * xk + B*u;
    xkp = xk + 0.5*h*K1; K2 = A * xkp + B * u;
    xkp = xk + 0.5*h*K2; K3 = A * xkp + B * u;
    xkp = xk + h*K2; K4 = A * xkp + B * u;
    xkp = xk + h/6*(K1 + 2*K2 + 2*K3 + K4 );
    y(ih+1) = C * xkp;
    xk = xkp;
end
figure();
plot( t, y, 'r', 'linewidth',3);
ylabel('y');
grid;
hold on;
g1 = ss(A,B,C,D);
[sy, st]= step(g1);
plot(st,sy,'b');
legend('RK-4', 'Step');
```

仿真结果如图 5-2-2 所示。

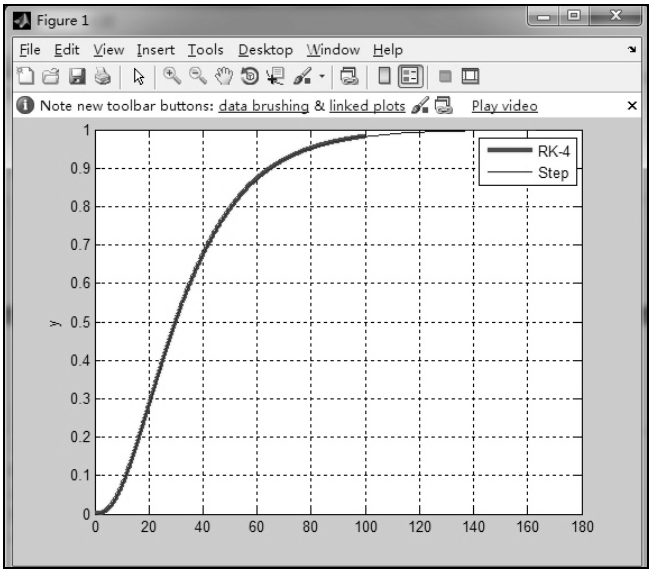


图 5-2-2　采用 RK-4 解法的曲线

（2）利用 Euler 法。程序如下：

```
num=[1];
den=conv(conv([5 1],[10 1]),[20 1]);
g=tf(num,den);
```

```
[A,B,C,D]=ssdata(g);
h = 1/50; u = 1;  x0 = [0; 0; 0]; xk=x0;
y(1) =C*xk+D*u; t(1) = 0;          %初始化
for(ih = 1:80/h)
    t(ih+1)=t(ih)+h;
    k1=A*xk+B*u;
    xkp=xk+h*k1;
    y(ih+1)=C*xkp+D*u;
    xk = xkp;
end
figure();
plot( t, y, 'r', 'linewidth',3);
ylabel('y');  grid;  hold on;
[sy, st]= step(g);
plot(st,sy,'b');
legend('eu', 'Step');
```

仿真结果如图 5-2-3 所示。

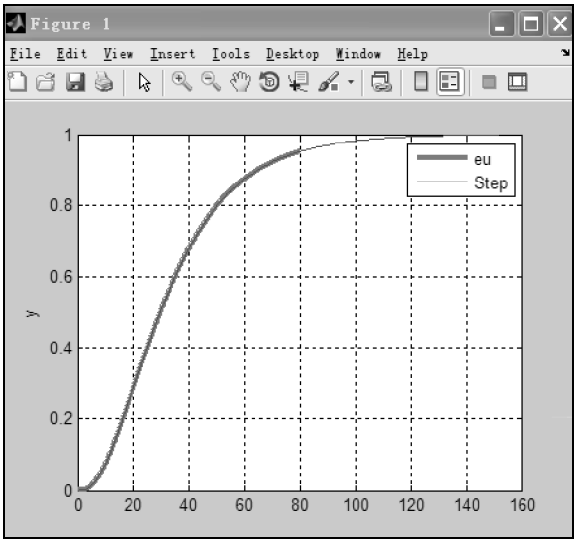


图 5-2-3　采用 Euler 法的曲线

2．离散相似法

离散相似法是将一个连续系统进行离散化处理，求得与其等价的离散模型。这种方法的实质是用常系数差分方程来等效原常系数微分方程，然后用迭代的方法求解差分方程。

连续系统可以用状态空间模型表示，因此可以基于状态方程离散化，得到时域离散相似模型。对传递函数作离散化处理得到离散传递函数，称为频域离散相似模型。

（1）时域离散相似基本原理。对连续系统进行离散化处理后得到离散相似模型（虚拟采样开关、虚拟信号重构器），根据离散状态方程得到与系统模型有关的系数：状态转移矩阵。

（2）频域离散相似法基本原理。将离散相似法用于连续传递函数从而得到系统离散传递函数。

5.2.3 离散事件系统仿真算法

离散事件系统的突出特点是系统性能状态变量只在随机的时间点上发生跃变，而在两个时间点之间不发生任何变化。这是这类系统区别于连续时间系统的主要方面，也是数学建模的根本依据。离散事件系统仿真不同于连续系统仿真，在连续系统仿真中，时间通常被分割成均等的或非均等的间隔，并以一个基本的时间计时，而离散系统的仿真则经常是面向事件的，系统中的状态只是在离散时间点上发生变化，而且这些离散时间点一般是不确定的，如理发馆系统、订票系统、库存系统、交通控制系统等。

在连续系统中，仿真结果表现为系统变量随时间变化的时间历程；在离散事件系统仿真中，系统变量是反映系统各部分相互作用的一些事件，仿真结果是产生处理这些事件的事件历程。离散事件的仿真，可采用三种方法。

1．事件调度法

按这种方法建立模型时，所有事件均放于事件表中，模型中设有一个时间控制成分，该成分从事件表中选择具有最早发生时间的事件，并将仿真中的时间变量，即在仿真中修改到该事件发生的时间，再调用与该事件相应的事件处理模块，该事件处理完毕后返回时间控制成分。这样，事件的选择与处理不断地进行，直到仿真终止的程序事件发生为止。在这种方法中，任何条件的测试，均在相应的事件模块中进行，是一种面向时间的仿真算法。

2．活动扫描法

在这类仿真中，系统由部件组成，而部件包含着活动，该活动是否发生，视规定的条件是否满足而定，因而有一专门模块来确定激活条件。若条件满足，则激活相应部件的活动模块。时间控制程序较其他条件具有更高的优先级，即在判断激活条件时首先判断该活动发生的时间是否满足，然后再判断其他条件。若所有条件都满足，则执行该部件的活动子程序。然后再对其他部件进行扫描，对所有部件扫描一遍后，又按同样顺序进行循环扫描，直到仿真终止。

3．进程交互法

这种方法综合了事件调度法和活动扫描法这两种方法的特点。采用两张事件表，即当前事件表与将来事件表。当前事件表包含了从当前时间点开始有资格执行的事件记录。每一个事件记录中包含该事件的若干属性，其中必有一个属性，说明该事件在过程中所处位置指针。

进程交互法首先按一定分布产生到达实体并置于将来事件表中，实体进入排队等待，然后对当前事件表进行扫描，判断各种条件是否满足，再对满足条件的活动进行处理，仿真软件推进到服务结束并将该实体从系统中清除，最后把将来事件表中为当前事件的实体移到当前事件表中。

5.2.4 系统的稳定性与仿真精度

1．系统的稳定性

采用计算机对系统进行动态仿真，首先要考虑系统的稳定性，只有稳定的系统其仿真结果才有意义。但有时会出现这种情况，即原本稳定的系统，经仿真计算后的数值却是发散的，这属于计算稳定性不符合要求。由于系统仿真时存在误差，对仿真结果会产生影响，若计算结果对系统仿真的计算误差反应不敏感，那么称之为算法稳定，否则称算法不稳定。对于不稳定的算法，设差会不断积累，最终可能导致仿真计算达不到系统要求而失败。

例如，计算

$$I_n = \mathrm{e}^{-1}\int_0^1 x^n \mathrm{e}^x \mathrm{d}x, \qquad n = 0,\ 1,\ \cdots$$

由分部积分得递推公式：

$$I_n = 1 - nI_{n-1}, \qquad I_0 = \mathrm{e}^{-1}\int_0^1 \mathrm{e}^x \mathrm{d}x = 1 - \mathrm{e}^{-1}$$

用 Taylor 展开计算：

$$\mathrm{e}^{-1} \approx 1 + (-1) + \frac{(-1)^2}{2!} + \cdots + \frac{(-1)^k}{k!} \approx 0.3679$$

依此方法逐步推算 10 个数值，发现积分值已出现负值，显然算法有问题。

分析计算误差：

$$E_n = I_n - \tilde{I}_n$$

满足关系：

$$E_n = -nE_{n-1} \qquad E_n = (-1)^n (n!) E_0$$

此算法误差增长迅速。如果换一种算法，可以减小误差，考虑积分估计值：

$$\frac{\mathrm{e}^{-1}}{n+1} = \mathrm{e}^{-1}\left(\min_{0\leqslant x\leqslant 1} \mathrm{e}^x\right)\int_0^1 x^n \mathrm{d}x < I_n < \mathrm{e}^{-1}\left(\max_{0\leqslant x\leqslant 1} \mathrm{e}^x\right)\int_0^1 x^n \mathrm{d}x \frac{1}{n+1}$$

$$I_{n-1}^* = \frac{1}{n}(1 - I_n^*) \qquad E_{n-1} = -\frac{1}{n}E_n$$

以此方法进行逆向计算，分析计算误差，显然误差一直减小，可满足计算要求。

2．仿真过程的三类误差

系统仿真很注重仿真精度，要达到用户规定的精度要求，就应当分析仿真中存在的误差及产生误差的原因，找到适当的方法加以改进。

数值积分法求解微分方程，实质上是通过差分方程作为递推公式进行的，在计算机逐次计算时，初始数据的误差及计算过程的舍入误差等都会使误差不断积累。如果这种仿真误差积累能够抑制，不会随计算时间增加而无限增大，则认为相应的计算方法是数值稳定的。反之则是数值不稳定的。

仿真误差与数值计算方法、计算机的精度以及计算步长的选择有关。当计算方法和计算机确定以后，则仅与计算步长有关。仿真误差一般有如下三种。

1）初始误差

在对系统仿真时，要采集现场的原始数据，而计算时要提供初始条件，这样由于数据的采集不一定很准，会造成仿真过程中产生一定的误差，此类误差称为初始误差。要消除或减小初始误差，就应对现场数据进行精确的检测，也可以多次采集，以其平均值作为参考初始数据。

2）截断误差

当仿真步长确定后，采用的数值积分公式的阶次将导致系统仿真时产生截断误差，阶次越高，截断误差越小。通常仿真时多采用四阶龙格-库塔法，其原因就是这种计算公式的截断误差较小。通常计算步长越小，截断误差也越小。

3）舍入误差

由计算机的精度有限（有限位数）所产生。由于计算机的字长有限，不同档次的计算机其计算结果的有效值不一致，导致仿真过程出现舍入误差。一般情况下，要降低舍入误差应选择档次高些的计算机，其字长越长，仿真数值结果尾数的舍入误差就越小。此外，计算步长越小，计算次数越多，舍入误差愈大。

3．计算步长的选择

一个数值解是否稳定，取决于该系统微分方程的特征根是否满足稳定性要求，而不同的数值积分公式具有不同的稳定区域，在仿真时要保证稳定就要合理选择仿真步长，使微分方程的解处于稳定区域之中。

对截断误差而言，计算步长越小越好，但太小不但会增加计算时间，而且由于舍入误差的增加，不一定能达到提高精度的目的，甚至可能出现数值不稳情况。计算步长太大，精度又不能满足要求，而且计算步长超过该算法的判稳条件时，也会出现不稳定情况。由此可见。计算步长只能在某一范围内选择。

一般控制系统的输出动态响应在开始变化较快，到最后变化将会很缓慢。这时，计算可以用变步长的方法，即在开始阶段步长取得小一些，在最后阶段取得大一些，这样即可以保证计算的精度，也可以加快计算的速度。

对于一般工程计算，如果计算精度要求不太高。可采用定步长的方法。

由于积分步长直接与系统的仿真精度和稳定性密切相关，所以，应合理地选择积分步长的值，以保证仿真结果符合用户要求。通常，积分步长 h 的选择要遵循以下两个原则：

（1）保证仿真系统的算法稳定。当已知系统最小时间常数 t 时，根据理论推导所得到的经验公式，采用欧拉法仿真要选择 $h \leqslant 2T$，采用四阶龙格-库塔法仿真应选择 $h \leqslant 5.78T$。

（2）保证仿真系统具备一定的计算精度。一般情况下，仿真中的初始误差及舍入误差对仿真过程影响。

在实际工程应用中常应用局部截断误差分析，整体阶段误差分析，绝对稳定分析等概念和性质来分析数值解法。

第 6 章　动态建模工具 Simulink

Simulink 是 MATLAB 用来对动态系统进行建模、仿真和分析的软件包，是 MATLAB 相对独立的重要组成部分。Simulink 可以很方便地创建和维护线性和非线性、连续和离散或者两者的混合系统，评估不同的算法和结构并验证系统的性能。由于 Simulink 是采用模块组合方式来建模，从而可以使得用户能够快速、准确地创建动态系统的计算机仿真模型，特别是对于复杂的不确定非线性系统，这种模型可读性强，可避免编程的繁琐，使得一个复杂系统模型的建立和仿真变得相当简单和直观。

6.1　Simulink 概述

Simulink 是 SIMU（仿真）和 LINK（链接）的合写，Simulink 的使用对象广泛，既支持连续与离散系统以及连续离散混合系统，也支持线性与非线性系统，还支持具有多种采样频率的系统，即不同的系统能够以不同的采样频率进行组合，以仿真复杂的系统。

Simulink 中不但实现了可视化的动态仿真，也实现了与 MATLAB、C 或者 FORTRAN 甚至和硬件之间的相互数据传递，大大扩展了它的功能。

Simulink 还提供一套图形动画的处理方法，使用户可以方便的观察到仿真的整个过程。

6.1.1　操作环境

在 MATLAB 的命令窗口中输入 Simulink，或单击 MATLAB 主窗口的快捷按钮 ，进入 Simulink。Simulink 环境主要由两大部分组成。

1．自身包含的一些系统建模的模型库

为便于用户能够快速构建自己所需的动态系统，Simulink 提供了大量以图形方式给出的内置系统模块，使用这些内置模块可以快速方便地设计出特定的动态系统。Simulink 的模块库能够对系统模块进行有效的管理与组织，使用 Simulink 模块库浏览器可以按照类型选择合适的系统模块、获得系统模块的简单描述以及查找系统模块等，并可以直接将模块库中的模块拖动或者复制到用户的系统模型中以构建动态系统模型。所有的库都能进行扩充和定制。Simulink 的开放式结构允许用户扩展仿真环境的功能：采用 MATLAB、FORTRAN 和 C 代码生成自定义模块库，并拥有自己的图标和界面。Simulink 还提供了很多有趣的演示程序。

2．MATLAB 工具箱的集成

MATLAB 自身所带的所有应用工具箱，同样适用于 Simulink 环境中。用户可以直接利用涉及众多领域的 MATLAB 工具来创建、分析和优化系统，用于处理某些特殊类型的问题。工具箱并不仅仅是一些有用函数的集合，在某种意义上它们代表着世界顶级研究人员在各自领域所做出的努力。也因为如此，随着 MATLAB 版本的升级，大大扩展了 Simulink 的适用范围，也使 Simulink 进入到许多专业领域，成为科学计算和动态系统仿真的有利工具，Simulink 与 MATLAB 其他组件的关系如图 6-1-1 所示。

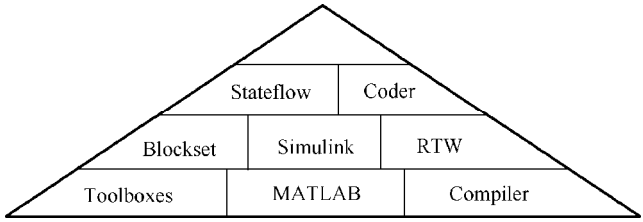


图 6-1-1　Simulink 在 MATLAB 家族中的位置

6.1.2　运行原理

1．仿真的特点

Simulink 仿真的特点有

- 外表：直观的方框图；
- 文件：MDL 文件；
- 数学：微分方程或差分方程；
- 行为：模拟物理过程的动态性状。

2．仿真过程

Simulink 对系统仿真的控制是通过系统模型与求解器之间建立对话的方式进行的：Simulink 将系统模型、模块参数与系统方程传递给 Simulink 的求解器，而求解器将计算出的系统状态与仿真时间通过 Simulink 环境传递给系统模型本身，通过这样的交互作用方式来完成动态系统的仿真。

Simulink 仿真包括两个阶段：初始化阶段和模型执行阶段。

1）*初始化阶段*

（1）对模型的参数进行估计，得到它们实际计算的值；
（2）展开模型的各个层次；
（3）按照更新的次序对模型进行排序；
（4）确定显式化的信号属性，并检查每个模块是否能够接收连接它们输入端的信号；
（5）确定所有非显式的信号采样时间模块的采样时间；

（6）分配和初始化存储空间，以便存储每个模块的状态和当前值的输出。

2）模型执行阶段

模型仿真是通过数值积分来进行完成的，计算数值积分可以采用以下两步来进行。

（1）按照秩序计算每个模块的积分；

（2）根据当前输入和状态来决定状态的微分，得到微分矢量，然后把它返回给解法器，以计算下一个采样点的状态矢量。在每一个时间步中，Simulink 依次解决下列问题。

① 按照秩序更新模块的输出。Simulink 计算一个模块的离散状态的方法时调用模块的离散状态更新函数。而对于连续状态，则对连续状态的微分（在模块可调用的函数里，有一个用于计算连续微分的函数）进行数值积分来获得当前的连续状态。

② 按照秩序更新模块的状态。Simulink 计算一个模块的离散状态的方法时调用模块的离散状态更新函数。而对于连续状态，则对连续状态的微分（在模块可调用的函数里，有一个用于计算连续微分的函数）进行数值积分来获得当前的连续状态。

③ 检查模块连续状态的不连续点。Simulink 使用过零检测来检测连续状态的不连续点。

④ 计算下一个仿真时间步的时间。通过调用模块获得下一个采样时间函数来完成。

6.2 模块库及模块功能

模块库提供了各种基本模块，按应用领域以及功能组成若干子库，并按树状结构进行显示，模块是 Simulink 建模的基本元素。

Simulink 模块库包括公共模块库和专业模块库两类。

6.2.1 Simulink 公共模块库

Simulink 公共模块库是 Simulink 中最为基础、最为通用的模块库，它可以被应用到不同的专业领域中。Simulink 公共模块库共包含 Continuous（连续模块库）、Discontinuities（非线性模块库）、Discrete（离散模块库）、Logic and Bit Operations（逻辑与位操作模块库）、Lookup Tables（查询表模块库）、Math Operations（数学运算模块库）、Model Verification（模型验证模块库）、Model-wide utilities（模块实用模块库）、Ports & Subsystems（端口和子系统模块库）、Signal Attributes（信号属性模块库）、Signal Routing（信号路由模块库）、Sinks（接收器模块库）、Sources（输入源模块库）、User-defined functions（用户自定义模块库）等模块库，公共模块库窗口如图 6-2-1 所示。

1．Continuous（连续系统）

Continuous 由描述标准线性函数和线性系统的模块组成，如表 6-2-1 所示。

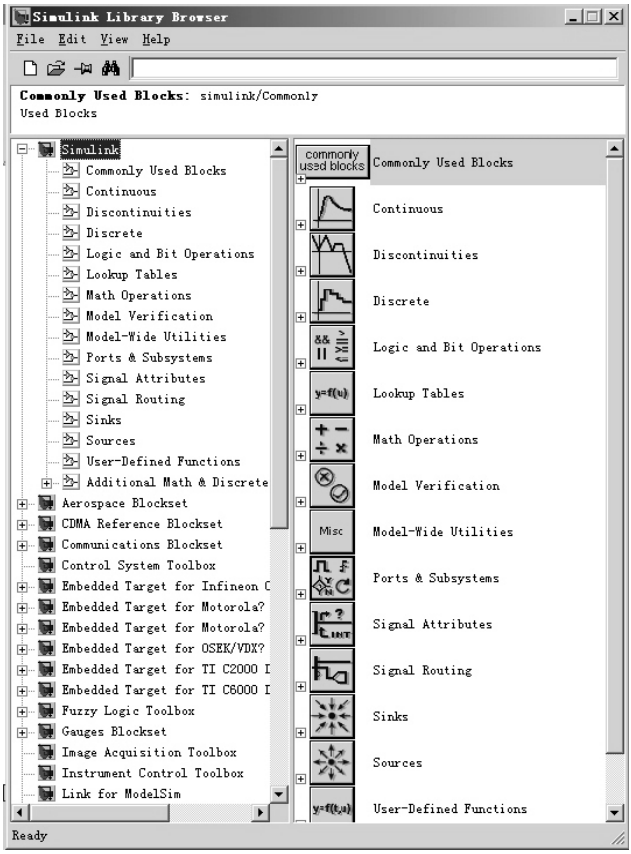


图 6-2-1　Simulink 的公共模块库

表 6-2-1　连续系统模块

名　　称	功 能 说 明	名　　称	功 能 说 明
Derivative	微分环节	Integrator	积分环节
State-Space	状态方程模型	Transfer Fcn	传递函数模型
Transport Delay	以给定的时间量延迟输入	Variable Transport Delay	以可变的时间量延迟输入
Zero-Pole	零-极点增益模型	—	—

2．Discontinuities（非连续系统）

Discontinuities 由描述非线性函数和非线性系统的模块组成，如表 6-2-2 所示。

表 6-2-2　非连续系统模块

名　　称	功 能 说 明
Backlash	间隙非线性（模拟有间隙系统的行为）
Coulomb&Viscous Friction	库仑和黏度摩擦非线性（模拟在零点处不连续，在其他地方有线性增益的系统）
Dead Zone	死区非线性（提供输出为零的区域）
Dead Zone Dynamic	动态死区非线性（动态提供输出为零的区域）
Hit Crossing	冲击非线性（检测信号上升沿、下降沿以及与指定值的比较结果，输出 0 或 1）

（续表）

名　　称	功 能 说 明
Quantizer	量化非线性（以指定的间隔离散化输入）
Rate Limiter	静态限制信号的变化速率（以指定的间隔离散化输入）
Rate Limiter Dynamic	动态限制信号的变化速率（限制信号的变化速度）
Relay	滞环比较器，限制输出值在某一范围内变化（在两个常数中选出一个作为输出）
Saturation	饱和输出，让输出超过某一值时能够饱和（限制信号的变化范围）
Saturation Dynamic	动态饱和输出（动态限制信号的变化范围）
Wrap To Zero	还零非线性（输入大于门限则输出零，小于则直接输出）

3．Discrete（离散系统）

Discrete 由描述离散时间系统的模块组成，如表 6-2-3 所示。

表 6-2-3　离散系统模块 Discrete

名　　称	功 能 说 明	名　　称	功 能 说 明
Difference	差分环节	Discrete Derivative	离散微分环节
Discrete Filter	离散滤波器	Discrete State-Space	离散状态空间系统模型
Discrete Transfer-Fcn	离散传递函数模型	Discrete Zero-Pole	以零极点表示的离散传递函数模型
Discrete-time Integrator	离散时间积分器	First-Order Hold	一阶保持器
Integer Delay	整数被延迟	Memory	输出本模块上一步的输入值
Tapped Delay	延迟 N 个周期，然后输出所有延迟数据	Transfer Fcn First Order	离散一阶传递函数
Transfer Fcn Lead or Lag	传递函数	Transfer Fcn Real Zero	离散零点传递函数
Unit Delay	一个采样周期的延迟	Weighted Moving Average	加权平均
Zero –Order Hold	零阶保持器	—	—

4．Logic and Bit Operations（逻辑和位操作模块）

Logic and Bit Operations 包括逻辑操作及位操作等模块，如表 6-2-4 所示。

表 6-2-4　逻辑和位操作模块

名　　称	功 能 说 明	名　　称	功 能 说 明
Bit Clear	位清零	Bit Set	位置位
Bitwise Operator	逐位操作	Combinatorial Logic	组合逻辑
Compare To Constant	和常量比较	Compare To Zero	和零比较
Detect Change	检测跳变	Detect Decrease	检测递减
Detect Fall Negative	检测负下降沿	Detect Fall Nonpositive	检测非负下降沿
Detect Increase	检测递增	Detect Rise Nonnegative	检测非负上升沿
Detect Rise Positive	检测正上升沿	Extract Bits	提取位
Interval Test	检测开区间	Interval Test Dynamic	动态检测开区间
Logical Operator	逻辑操作符	Relational Operator	关系操作符
Shift Arithmetic	移位运算	—	—

5．Lookup Tables（查找表）

Lookup Tables 包括各种查表和插值模块，如表 6-2-5 所示。

表 6-2-5　查表模块

名　　称	功 能 说 明
Cosine	余弦函数查询表
Direct Lookup Table (n-D)	*N* 个输入信号的查询表（直接匹配）
Interpolation(n-D) using PreLookup	*N* 个输入信号的预插值
Lookup Table	输入信号的查询表（线性峰值匹配）
Lookup Table(2-D)	两维输入信号的查询表（线性峰值匹配）
Lookup Table(n-D)	*N* 维输入信号的查询表（线性峰值匹配）
Lookup Table Dynamic	动态查询表
PreLookup Index Search	预查询索引搜索
Sine	正弦函数查询表

6．Math Operations（数学计算）

Math Operations 由描述数学运算的模块组成，包括数学运算、向量运算、复数与向量间的转换运算，如表 6-2-6 所示。

表 6-2-6　数学运算模块

名　　称	功 能 说 明	名　　称	功 能 说 明
Abs	取绝对值	Add	加法（或减法）
Algebraic Constraint	代数约束（将输入信号抑制为零）	Assignment	赋值
Bias	偏移（给输入加入偏移量）	Complex to Magnitude-Angle	由复数输入转为幅值和相角输出
Complex to Real-Imag	由复数输入转为实部和虚部输出	Divide	除法
Dot Product	点乘运算	Gain	比例运算
Magnitude-Angle to Complex	由幅值和相角输入合成复数输出	Math Function	常用数学函数
Matrix Concatenation	矩阵级联	MinMax	最值运算
MinMax Running Resettable	最大最小值运算(带复位功能)	Polynomial	多项式
Product	乘运算	Product of Elements	元素乘运算
Real-Imag to Complex	由实部和虚部输入合成复数输出	Reshape	改变矩阵或向量的维数
Rounding Function	舍入函数	Sign	符号函数（指明输入的符号）
Sine Wave Function	正弦波函数	Slider Gain	滑动增益
Subtract	对信号进行加法或减法运算	Sum	求和运算
Sum of Elements	元素和运算	Trigonometric Function	三角函数
Unary Minus	一元减法（对输入取反）	Weighted Sample Time Math	权值采样时间运算（对信号经过加权时间采样的值进行加、减、乘、除运算）

7．Model Verification（模型检测）

Model Verification 包括仿真建模中各种特性的检测模块，如表 6-2-7 所示。

表 6-2-7　模型检测模块

名　称	功 能 说 明	名　称	功 能 说 明
Assertion	确定操作	Check Discrete Gradient	检查离散梯度
Check　Dynamic Gap	检查动态偏差	Check Dynamic　Lower Bound	检查动态下限
Check　Dynamic Range	检查动态范围	Check Dynamic　Upper Bound	检查动态上限
Check Input　Resolution	检查输入精度	Check　Static Gap	检查静态偏差
Check Static　Lower Bound	检查静态下限	Check　Static Range	检查静态范围
Check Static　Upper Bound	检查静态上限	—	—

8．Model-Wide Utilities（模型扩充）

Model-Wide Utilities 如表 6-2-8 所示。

表 6-2-8　模型扩充模块

名　称	功 能 说 明	名　称	功 能 说 明
Block Support Table	功能块支持的表	DocBlock	文档模块
Model Info	模型信息	Timed-Based inearization	时间线性分析
Trigger-Based Linearization	触发线性分析	—	—

9．Ports&Subsystems（端口和子系统）

Ports&Subsystems 包括各种类似程序结构和子系统模块，如表 6-2-9 所示。

表 6-2-9　端口和子系统模块

名　称	功 能 说 明	名　称	功 能 说 明
Configurable Subsystem	结构子系统	Atomic Subsystem	单元子系统
CodeReuseSubsystem	代码重用子系统	Enable	使能
Enabled and Triggered Subsystem	使能和触发子系统	Enabled Subsystem	使能子系统
For Iterator Subsystem	重复操作子系统	Function-Call Generator	函数响应生成器
Function-Call Subsystem	函数响应子系统	If	假设操作
If Action Subsystem	假设动作子系统	In1	输入端口
Model	模型	Out1	输出端口
Subsystem	子系统	Subsystem Examples	子系统例子
Switch Case	转换事件	Switch Case Action Subsystem	转换事件子系统
Trigger	触发操作	Triggered Subsystem	触发子系统
While Iterator Subsystem	重复子系统	—	—

10．Signal Attributes（信号属性）

Signal Attributes 包括对不同信号数据类型的处理和匹配的模块，如表 6-2-10 所示。

表 6-2-10　信号属性模块

名　　称	功 能 说 明	名　　称	功 能 说 明
Data Type Conversion	数据类型转换	Data Type Conversion Inherited	继承的数据类型转换
Data Type Duplicate	数据类型复制	Data Type Propagation	数据类型继承
Data Type Propagation Examples	数据类型继承例子	Data Type Scaling Strip	数据类型缩放
IC	信号输入属性	Probe	探针点
Rate Transition	比率转换	Signal Conversion	信号转换
Signal Specification	信号特征说明	Weighted Sample Time	权值采样时间
Width	信号宽度	—	—

11．Signal Routing（信号线路）

Signal Routing 包括信号传输与信号存储、访问的模块，如表 6-2-11 所示。

表 6-2-11　信号线路模块

名　　称	功 能 说 明	名　　称	功 能 说 明
Bus Assignment	总线分配	Bus Creator	总线生成
Bus Selector	总线选择	Data Store Memory	数据存储
Data Store Read	数据存储读取	Data Store Write	数据存储写入
Environment Controller	环境控制器	From	信号来源
Goto	信号去向	Goto Tag Visibility	标签可视化
Index Vector	索引向量	Manual Switch	手动选择开关
Merge	信号合并	Multiport Switch	多端口开关
Mux	将多个单一输入转化为一个复合输出	Demux	将一个复合输入转化为多个单一输出
Selector	信号选择器	Switch	开关选择，当第二个输入端大于临界值时，输出由第一个输入端而来，否则输出由第三个输入端而来

12．Sinks（接收器）

Sinks 包含模型及子系统输出、数据观察器与仿真控制的模块，如表 6-2-12 所示。

表 6-2-12　接收器模块

名　　称	功 能 说 明	名　　称	功 能 说 明
Display	数字显示器	Floating Scope	浮动观察器
Out1	输出端口	Scope	示波器
Stop Simulation	仿真停止	Terminator	连接到没有连接到的输出端
To File(.mat)	将输出数据写入数据文件保护	To Workspace	将输出数据写入 MATLAB 的工作空间
XY Graph	显示二维图形	—	—

13．Sources（输入源）

Sources 包含多种常用的信号和数据发生器及子系统输入模块，如表 6-2-13 所示。

表 6-2-13　输入源模块

名　　称	功 能 说 明
Band-Limited White Noise	带限白噪声
Chirp Signal	产生一个频率递增的正弦波（线性调频信号）
Clock	显示和提供仿真时间
Constant	常数信号
Counter Free-Running	自运行计数器，计数溢出时自动清零
Counter Limited	有限计数器，可自定义计数上限
Digital Clock	在规定的采样间隔产生仿真时间
From File(.mat)	从文件读取数据
From Workspace	从工作空间中定义的矩阵中读取数据
Ground	地线，提供零电平
In1	提供一个输入端口
Pulse Generator	脉冲发生器
Ramp	斜坡输入
Random Number	产生正态分布的随机数
Repeating Sequence	产生规律重复的任意信号
Repeating Sequence Interpolated	重复序列内插值
Repeating Sequence Stair	重复阶梯序列
Signal Builder	信号创建器
Signal Generator	信号发生器，可产生正弦、方波、锯齿波及随意波
Sine Wave	正弦波信号
Step	阶跃信号
Uniform Random Number	生成均匀分布的随机数

14．User-Defined Functions（用户自定义函数）

User-Defined Functions 包括用户自定义的函数模块，如表 6-2-14 所示。

表 6-2-14　用户自定义函数模块

名　　称	功 能 说 明
Embedded MATLAB Function	嵌入的 MATLAB 函数
Fcn	用自定义的函数（表达式）进行运算
M-file S-Function	M 文件编写的 S 函数
MATLAB Fcn	利用 MATLAB 的现有函数进行运算
S-Function	调用自编的 S 函数的程序进行运算
S-Function Builder	S 函数建立器
S-Function Examples	S 函数例子

6.2.2　Simulink 专业模块库

所谓专业模块库，实际上就是模块集及常用的工具箱，是 Simulink 主库中的独立程序库的集合。在此集成环境下，Simulink 不再是单一的动态系统仿真软件，它的作用和应用领域已经得到进一步的扩展。

这里介绍常用专业模块库的主要功能。

（1）Control system TooIbox 模块库：面向控制系统的设计与分析，主要提供线性时不变系统的模块。

（2）DSP Blockset 模块库：面向数字信号处理系统的设计与分析，主要提供 DSP 输入模块、DSP 输出模块、信号预测与估计模块、滤波器模块、DSP 数学函数库、量化器模块、信号管理模块、信号操作模块、统计模块以及信号变换模块等。

（3）Simulink Extras 模块库：主要补充 Simulink 公共模块库，提供附加连续模块库、附加线性系统模块库、附加输出模块库、触发器模块库、线性化模块库、系统转换模块库以及航空航天系统模块库等。

（4）S-funcitiong demos 模块库：主要提供 C++、C、FORTRAN 以及 M 文件下 S-函数的模块库的演示模块。

（5）Real Time workshop 与 Real-Time Windows Target 模块库：主要提供各种用来进行独立可执行代码或嵌入式代码生成，以实现高效实时仿真的模块，它们和 RTW、TLC 有着密切的联系。

（6）Stateflow 库：对使用状态图所表达的有限状态机模型进行建模仿真和代码生成，有限状态机用来描述基于事件的控制逻辑，也可用于描述响应型系统。

（7）Fix-point blockset：包含一组用于定点算法仿真的模块。

（8）Communication blockset：专用于通信系统仿真的一组模块。

（9）Dials＆Gauges 库：图形仪表模块库，它们实际上是一组 ActiveX 控件。

（10）Neural Network blockset：用于神经网络的分析设计和实现的一组模块。

（11）Fuzzy logic toolbox：包括一组有关模糊控制的分析设计和实现的模块。

（12）xPC 模块：提供了一组用于 xPC 仿真的模块。所谓的 xPC 是指利用 PC，使用客户端服务器的模式进行实时仿真的一种经济仿真方案，它和 Simulink、RTW 相结合，可以在 PC 下进行单任务的实时仿真。

6.3 建模及仿真

利用 Simulink 进行系统建模和仿真的一般步骤为：

（1）绘制系统流图；

（2）启动 Simulink 模块库浏览器，新建一个空白模型窗口；

（3）将所需模块放入空白模型窗口中，按系统流图的布局连接各模块，并封装子系统；

（4）设置各模块的参数以及与仿真有关的各种参数；

（5）保存模型，模型文件的后缀名为“.mdl”；

（6）运行并调试模型。

6.3.1 模块的基本操作

模块的操作主要包括以下一些基本操作，其方法比较简单，一般均可以采用直接拖曳或用鼠标右键菜单进行选取相应功能以及窗口菜单上选择相应菜单项操作即可实现。常用的模块操作有：

- 模块的提取；
- 模块的移动、放大和缩小；
- 模块的复制和粘贴；
- 模块的删除和恢复；
- 模块的转向；
- 模块名的修改和移动；
- 模块颜色的改变；
- 模块的参数设置；
- 模块的属性设定；
- 模块的连接；
- 连线的弯折、移动和删除；
- 批处理方法。

Simulink 中几乎所有模块的参数都允许用户进行设置，只要双击要设置的模块或在模块上按鼠标右键并在弹出的菜单中选择 Block Parameters 就会显示参数设置对话框。不同模块的对话框不同，每个对话框中有提示和帮助。

选中模块，打开 Edit 菜单的 Block Properties 也可以对模块进行属性设定，包括 Description 属性、Priority 优先级属性、Tag 属性、Open function 属性、Attributes formatstring 属性。其中 Open function 属性是一个很有用的属性，通过它指定一个函数名，则当该模块被双击之后，Simulink 就会调用该函数执行，这种函数在 MATLAB 中称为回调函数。

模块的连接包括改变粗细、设定标签、线的折弯、分支等。

模块处理的信号包括标量信号和向量信号；标量信号是一种单一信号，而向量信号为一种复合信号，是多个信号的集合，它对应着系统中几条连线的合成。默认情况下，大多数模块的输出都为标量信号，对于输入信号，模块都具有一种“智能”的识别功能，能自动进行匹配。某些模块通过对参数的设定，可以使模块输出向量信号。

模块连接的关键模块有 Mux，Demux 等。很多模块均支持向量化，如积分器等。

对于向量信号线，在“untitle”模型窗口里，选中主菜单项 Format 命令中的 Wide Vector Liners 命令，对模型执行完 Simulation 下的 Start 命令或 Edit 下的 Update Diagram 命令后，传输向量的线会变粗。此即说明变粗了的线段，表示该连接线上的信号为向量形式。

6.3.2 仿真参数设置

建立好系统模型后。就可以开始进行系统仿真。在仿真之前，可根据仿真系统的特性调整仿真参数。

选择 Simulink | Parameterss 选项设置仿真控制参数。参数设置标签页如图 6-3-1 所示。

仿真参数设置对话框主要包含以下选项卡。

（1）解法设置（Solver）：允许用户设置仿真的开始和结束时间，选择解法器，说明解法器参数及选择一些输出选项。

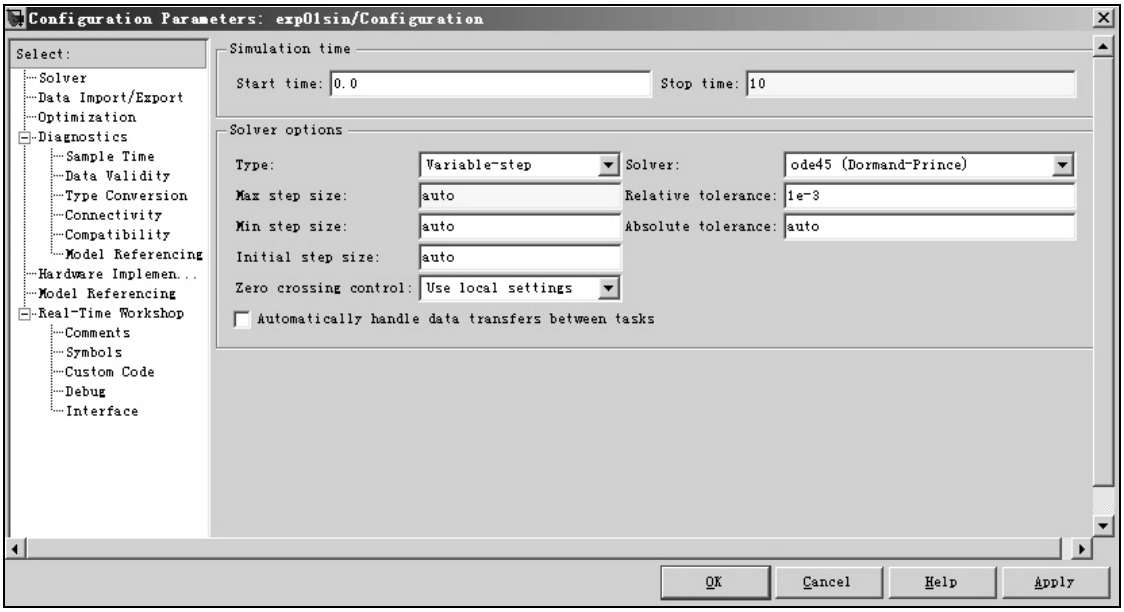


图 6-3-1　仿真参数设置及 Solver 标签

（2）数据的输入/输出（Data Import/Export）：允许用户从 MATLAB 的工作空间输入模型所需参数或将运行结果输出到工作空间。

（3）优化选项（Optimization）：主要是对仿真性能进行优化设置。

（4）诊断项（Diagnostics）：允许用户选择 Simulink 在仿真中显示的警告信息的等级。

（5）实时工具对话框诊断项（Real-Time Workshop）：主要用于与 C 语言编辑器的交换，通过它可以直接从 Simulink 模型生成代码并且自动建立可以在不同环境下运行的程序，这些环境包括实时系统和单机仿真。

一般情况下，参数设置主要涉及到 Solver 选项，其中的参数设置直接影响到仿真模型是否能正常运行，下面详细介绍 Solver 选项卡的参数设置。

So1ver 标签如图 6-3-1 所示，其主要功能在于设置一些仿真过程中的基本参数，包括：

（1）仿真时间。Start time（起始时间）和 Stop time（终止时间），注意这里的时间概念与真实的时间并不一样，只是计算机仿真中对时间的一种表示，如 10 s 的仿真时间，如果采样步长定为 0.1，则需要执行 100 步，若把步长减小，则采样点数增加，那么实际的执行时间就会增加。

（2）仿真步长模式。可供选择的有 Variable-step（变步长）和 Fixed-step（固定步长）两种方式。变步长模式可以在仿真的过程中改变步长，提供误差控制和过零检测。用户还可以在第二个下拉选项框中选择对应模式下仿真所采用的算法。

① 步长参数：对于变步长模式，用户可以设置最大的和推荐的初始步长参数，默认情况下，步长自动地确定，它由值 auto 表示。

- Maximum step size（最大步长参数）：它决定了解法器能够使用的最大时间步长，它的默认值为“仿真时间/50”，即整个仿真过程中至少取 50 个取样点，但这样的

取法对于仿真时间较长的系统则可能带来取样点过于稀疏，而使仿真结果失真。一般建议对于仿真时间不超过 15 s 的采用默认值即可，对于超过 15 s 的每秒至少保证 5 个采样点，对于超过 100 s 的，每秒至少保证 3 个采样点。

- Initial step size（初始步长参数）：一般建议用“auto”默认值即可。

② 仿真精度的定义（对于变步长模式）。

- Relative tolerance（相对误差）：它是指误差相对于状态的值，是一个百分比，默认值为 1e-3，表示状态的计算值要精确到 0.1%。
- Absolute tolerance（绝对误差）：表示误差值的门限，或者是说在状态值为零的情况下，可以接受的误差。如果设成 auto，那么 Simulink 为每一个状态设置初始绝对误差为 1e-6。

（3）仿真算法设置。

① 变步长模式解法器有 ode45、ode23、ode113、ode15s、ode23s、ode23t、ode23tb 和 discrete。

- ode45：默认值，四/五阶龙格-库塔法，适用于大多数连续或离散系统，但不适用于刚性（stiff）系统。它是单步解法器，也就是，在计算 y(tn)时，它仅需要最近处理时刻的结果 y(tn-1)。一般来说，面对一个仿真问题最好是首先试试 ode45。
- ode23：二/三阶龙格-库塔法，它在误差限要求不高和求解的问题不太难的情况下，可能会比 ode45 更有效。也是一个单步解法器。
- ode113：是一种阶数可变的解法器，它在误差容许要求严格的情况下通常比 ode45 有效。ode113 是一种多步解法器，也就是在计算当前时刻输出时，它需要以前多个时刻的解。
- ode15s：是一种基于数字微分公式的解法器（NDFs），也是一种多步解法器。适用于刚性系统，当用户估计要解决的问题是比较困难的，或者不能使用 ode45，即使使用效果也不好，就可以用 ode15s。
- ode23s：是一种单步解法器，专门应用于刚性系统，在弱误差允许下的效果好于 ode15s，能解决某些 ode15s 所不能有效解决的刚性系统问题。
- ode23t：是梯形规则的一种自由插值实现。这种解法器适用于求解适度刚性系统的问题而用户又需要一个无数值振荡的解法器的情况。
- ode23tb：是 TR-BDF2 的一种实现，TR-BDF2 是具有两个阶段的隐式龙格-库塔公式。
- discrtet：当 Simulink 检查到模型没有连续状态时使用它。

② 固定步长模式解法器有 ode5、ode4、ode3、ode2、ode1 和 discrete。

- ode5：默认值，是 ode45 的固定步长版本，适用于大多数连续或离散系统，不适用于刚性系统。
- ode4：四阶龙格-库塔法，具有一定的计算精度。
- ode3：固定步长的二/三阶龙格-库塔法。
- ode2：改进的欧拉法。

- ode1：欧拉法。
- discrete：是一个实现积分的固定步长解法器，它适合于离散无连续状态的系统。

（4）任务和采样时间设置。通过参数设置来检查模块间的速率转换，在建立多任务系统模型时不同模型的采样时间可能不同步，通过此参数可进行调整。

6.3.3 Simulink 与 MATLAB 之间的数据交互

Simulink 与 MATLAB 之间可以通过多种方式进行数据交互。

1．用 MATLAB 工作空间变量设置系统模块参数

模块参数可以是常量也可以是工作空间中的变量或变量表达式。

2．将信号输出到 MATLAB 工作空间中

使用示波器模块 Scope 的输出信号，可以使用户对输出的信号进行简单的定性分析。使用 Sinks 模块库中的 To Workspace 模块，可以轻易地将信号输出到 MATLAB 工作空间中。信号输出的名称在 To Workspace 模块的对话框中设置，此对话框还可以设置输出数据的点数、输出的间隔，以及输出数据的类型等。其中输出类型有三种形式：数组、结构以及带有时间变量的结构。仿真结束或暂停时的信号被输出到工作空间中。

3．用工作空间变量作为系统输入信号

Simulink 与 MATLAB 的数据交互是相互的，除了可以将信号输出到 MATLAB 工作空间中之外，用户还可以使用 MATLAB 工作空间中的变量作为系统模型的输入信号。使用 Sources 模块库中的 From Workspace 模块可以将 MATLAB 工作空间中的变量作为系统模型的输入信号。

4．MATLAB Function 与 Function 模块

除了使用上述的方式进行 Simulink 与 MATLAB 之间的数据交互，用户还可以使用 Functions and Tables 模块库中的 Function 模块（简称 Fcn 模块）或 MATLAB Function 模块（简称 MATLAB Fcn 模块）进行彼此间的数据交互。

Fcn 模块一般用来实现简单的函数关系，在 Fcn 模块中：

- 输入总是表示成 u，u 可以是一个向量；
- 可以使用 C 语言表达式，例如 sin(u[1])+cos(u[2])，输出永远为一个标量。

MATLAB Fcn 一般用来调用 MATLAB 函数来实现一定的功能，在 MATLAB Fcn 模块中：

- 所要调用的函数只能有一个输出（可以是一个向量）；
- 单输入函数只需使用函数名，多输入函数输入需要引用相应的元素，如 mean、sqrt、myfunc(u(1),u(2))，在每个仿真步长内都需要调用 MATLAB 解释器。

图 6-3-2 所示的仿真图为一个分别使用 Fcn 模块和 MATLAB Fcn 模块进行数据交互的例子。

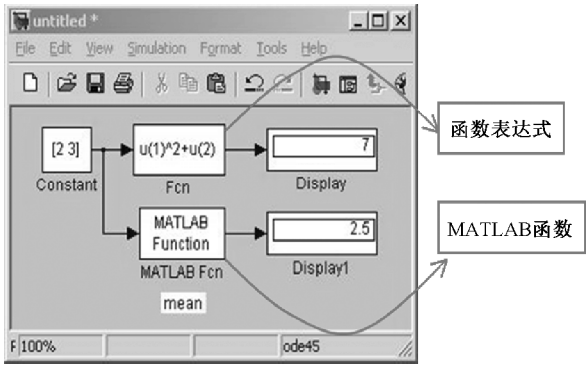


图 6-3-2　使用 Fcn 与 MATLAB Fcn 模块进行数据交互

6.3.4　模型的运行、结果观察和调试

1．模型的运行

建立好模型之后，就可以利用 Simulink 的菜单命令或者在 MATLAB 的命令窗口中输入命令进行仿真。

直接选择 Simulink 菜单项中的 start 运行仿真。在对控制系统进行仿真时，一般必须加入时钟信号，以给出仿真时间和便于使用变步长仿真。为了将仿真结果返回工作空间，还应该加上 To Workspace 模块，将输出和时间变量都返回。

注意在选择 To Workspace 模块参数时，输出向量的最大保存行数一定要与时间变量的最大保存行数保持一致，否则，就不能用 plot 函数在命令空间中画曲线。

在 MATLAB 命令窗口输入命令或用一个 M 文件也可以自动运行仿真。虽然使用菜单运行仿真十分简便，但从 MATLAB 命令窗口运行仿真可以在 M 文件中嵌入运行仿真程序，使得仿真和模块参数可通过程序交互式更改。

常用的命令仿真函数见表 6-3-1

表 6-3-1　Simulink 命令仿真函数

函　　数	意　　义
sim	运行一个由 Simulink 建立的模型
simset	为 sim 函数建立或编辑仿真参数或规定算法，并把设置结果保存在一个结构变量中
simget	获得模型的参数设置值
set_param	设置 Simulink 仿真参数

【例 6-3-1】利用命令方式对图 6-3-3 所示的模型进行仿真。

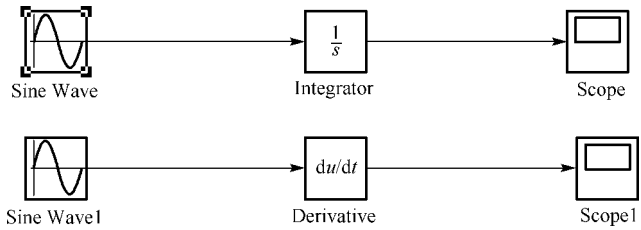


图 6-3-3　正弦信号的微分积分运算

将此模块命名为 sinsim.mdl。程序如下：

```
[t,x,y]=sim('sinsim');
[t,x,y]=sim('sinsim',[1,8]);
[t,x,y]=sim('sinsim',[2,4,6,8]);
option1=simset('outputvariables','xy','outputpoints','all');
[t,x,y]=sim('sinsim',[2,4,6,8],option1);
struct=simget('sinsim')
set_param('sinsim','starttime','10','stoptime','30')
set_param('sinsim','simulationcommand','start')
```

2．观察并分析仿真结果

MATLAB 软件规定，在 Simulink 仿真中必须要有输出显示环节，否则会报出错信息。在 Sinks 模块库中，提供了多种观察输出信号的方法。

- 示波器、浮动示波器输出；
- 直接数据显示；
- 输出端口；
- 返回工作空间；
- 文件输出；
- 表盘与量计显示。

示波器 Scope 模块是观察波形使用的最多的模块。Scope 模块将信号显示在其独立窗口中，以图形的方式直接显示指定的信号，在很多情况下，无须对输出结果进行定量分析，便可以从其仿真输出曲线中获知系统的运行规律。Scope 模块给用户提供了诸多控制手段，可以将数据保存到工作空间，可以使用户对模块的输出曲线进行控制调整，以便用户观测和分析输出结果。

Scope 模块的工具栏按钮命令如图 6-3-4 所示。

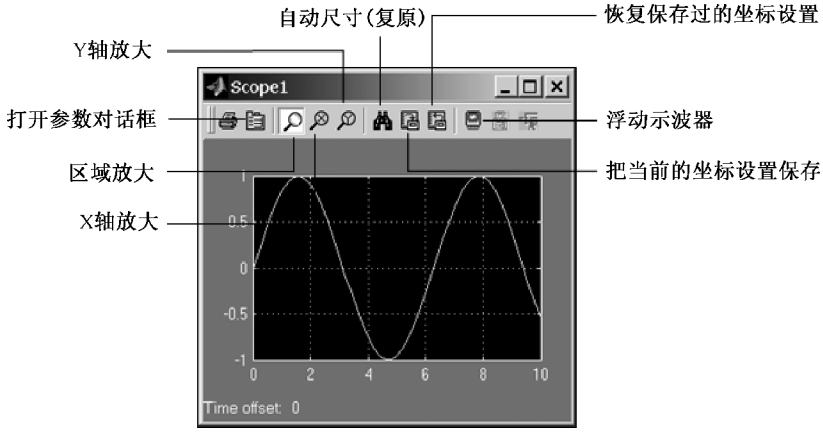


图 6-3-4　Scope 模块的工具栏按钮

此外，还可以使用悬浮 Scope 模块来观察仿真结果。在系统仿真分析中，用户往往需要利用多个输出信号进行观察分析。如果将每一个信号都与一个 Scope 模块相连接，则系

统模型必定会存在多个 Scope 模块，使得系统模型不够简练，而且难以对不同 Scope 模块中显示的信号进行直观比较。Sinks 模块库中的 Floating Scope（悬浮示波器）可以很好地解决这一问题。

3．仿真的调试技术

Simulink 提供了调试器，以方便查找和诊断模型中的错误，它允许通过单步运行仿真显示模块的即时状态、输入和输出。

Simulink 的图形调试器具有优秀的用户界面。使用菜单 Tools 下的 Simulink debugger 命令或使用调试器按钮可以启动调试器。图 6-3-5 所示为 Simulink 图形调试器窗口。

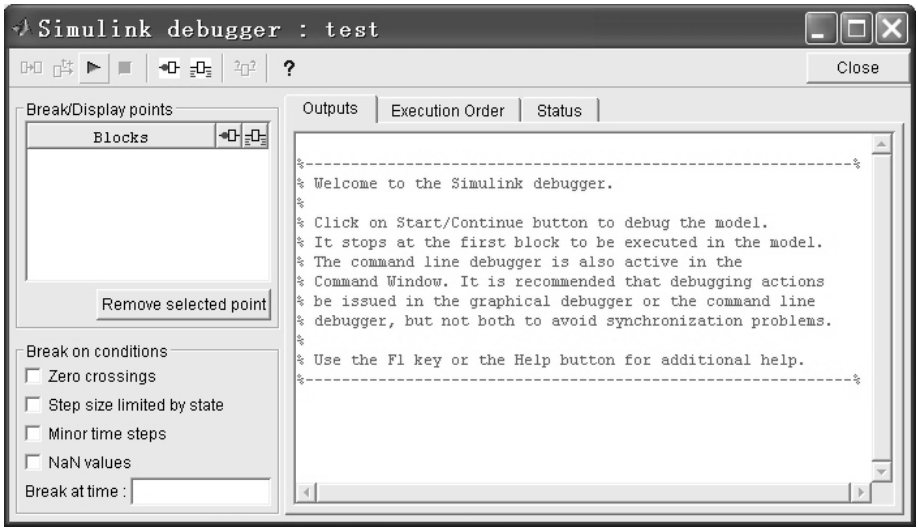


图 6-3-5　Simulink 图形调试器

启动 Simulink 调试器，设置合适的调试断点之后，便可以对系统模型中指定的模块或信号进行调试。

1）断点显示及断点条件设置

Simulink 提供了友好的调试界面，用户可以在断点显示框中了解到当前断点的信息，如断点位置、断点模块的输入输出等，一般来说，用户可以于调试前在指定的模块之前设置断点。但是多数情况下，用户需要在一定的条件下设置系统断点以进行调试。Simulink 调试器提供了五种断点条件设置，如图 6-3-6 所示。

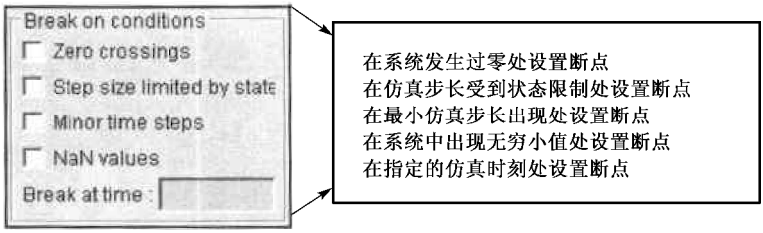


图 6-3-6　断点条件设置

2）调试器输出窗口

在对指定的系统模型进行调试时，调试结果均在 Simulink 的输出窗口显示，主要参数设置有

- Outputs：输出调试结果，如调试时刻、调试的模块以及模块输入输出等；
- Execution Order：输出调试顺序，即调试过程中各模块的执行顺序；
- status：输出调试状态，如当前仿真时间、默认调试命令（执行至下一模块或执行至下一时间步）、调试断点设置以及断点数等状态信息。

6.3.5 仿真实例

1．二进制编码器

这里所谓的二进制编码器即 8 线 3 线编码器，是指有 8 个信号输入端和 3 个输出端的编码器，其功能是对输入的 8 个信号进行编码，输出 3 个二进制数。

1）仿真分析

8 线 3 线编码器的真值表如表 6-3-2 所示。

表 6-3-2　8 线 3 线编码器真值表

输 入 信 号								输 出 信 号		
J0	J1	J2	J3	J4	J5	J6	J7	Y0	Y1	Y2
0	1	1	1	1	1	1	1	0	0	0
1	0	1	1	1	1	1	1	0	0	1
1	1	0	1	1	1	1	1	0	1	0
1	1	1	0	1	1	1	1	0	1	1
1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1

根据真值表写出的输入/输出间的逻辑函数为

$$\begin{cases} Y0 = \overline{J4 \cdot J5 \cdot J6 \cdot J7} \\ Y1 = \overline{J2 \cdot J3 \cdot J6 \cdot J7} \\ Y2 = \overline{J1 \cdot J3 \cdot J5 \cdot J7} \end{cases}$$

2）模块参数设置

首先按照前述方法建立新的模型窗口，然后将本次仿真需要的模块添加到模型中。本系统需要三种模块：

- 逻辑运算模块——与非门（3 个），用于实现编码器输入信号间的逻辑运算功能；
- 离散脉冲源（8 个），用于 8 个端口的脉冲信号输入；
- 示波器（2 个），用于显示输出的信号。

（1）逻辑模块参数设置。将参数 Operator 修改为“NAND（与非）”，Number of input ports 修改为 4。

（2）示波器模块参数设置。运算结果显示的示波器模块 Number of axes 修改为 3，显示输入信号的示波器坐标轴数目修改为 8。

（3）脉冲源。选择脉冲类型（Pulse type）为“基于采样（Sample based）”。接下来有 5 个参数需要设置。

- Amplitude：脉冲信号的幅度；
- Period：脉冲信号的周期（以样本数为单位）；
- Pulse width：脉冲宽度（即电平为 1 的时间，以样本数为单位）；
- Pulse delay：相位延迟（以样本数为单位）；
- Sample time：采样时间长度。

观察本例的真值表，注意到信号 J0～J7 的长度为 8，且 J0 到 J7 依次为低电平，所以将 J0 到 J7 的周期设为 8，脉冲宽度设为 7，相位延迟依次设为–7 到 0，脉冲幅度和采样时间使用默认值。这样在零时刻，J0 为低电平，其余输入为高电平；经过一个采样时间后，J1 变为低电平，如此持续下去，到第 7 个采样时间，J7 就变为低电平，实现了设计要求。

二进制编码器仿真模型如图 6-3-7 所示。

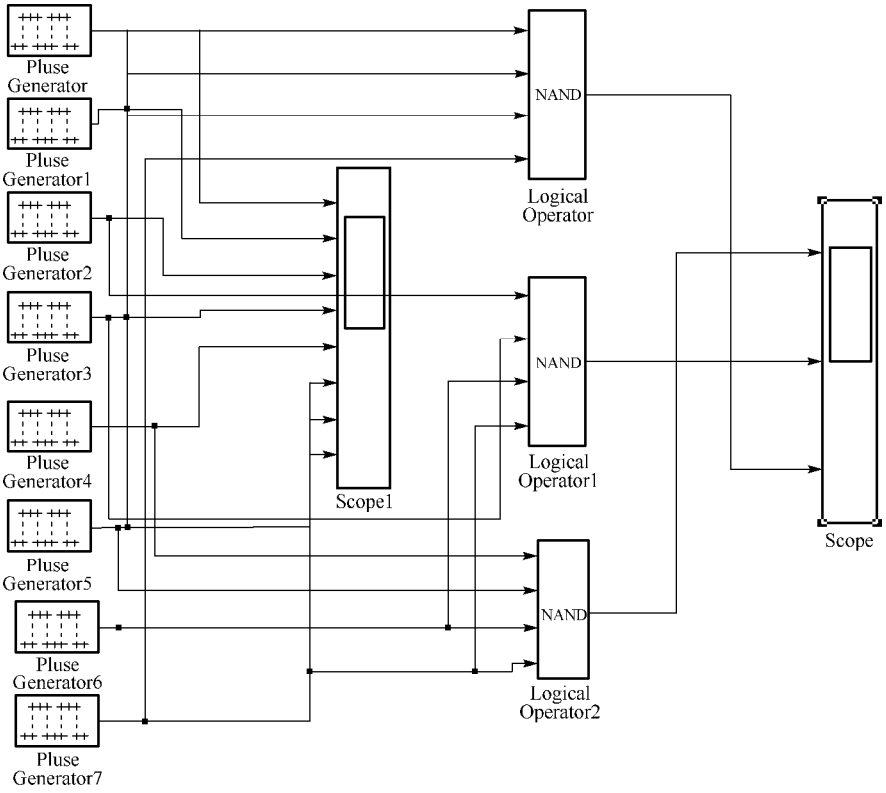


图 6-3-7　二进制编码器仿真模型

连接完成后，即可运行仿真（仿真参数采用默认设置即可）。仿真结束后，双击 Scope 观察波形结果，如图 6-3-8 所示，J0～J7 的输入波形，J0～J7 以 8 为周期，依次出现 0 电平。

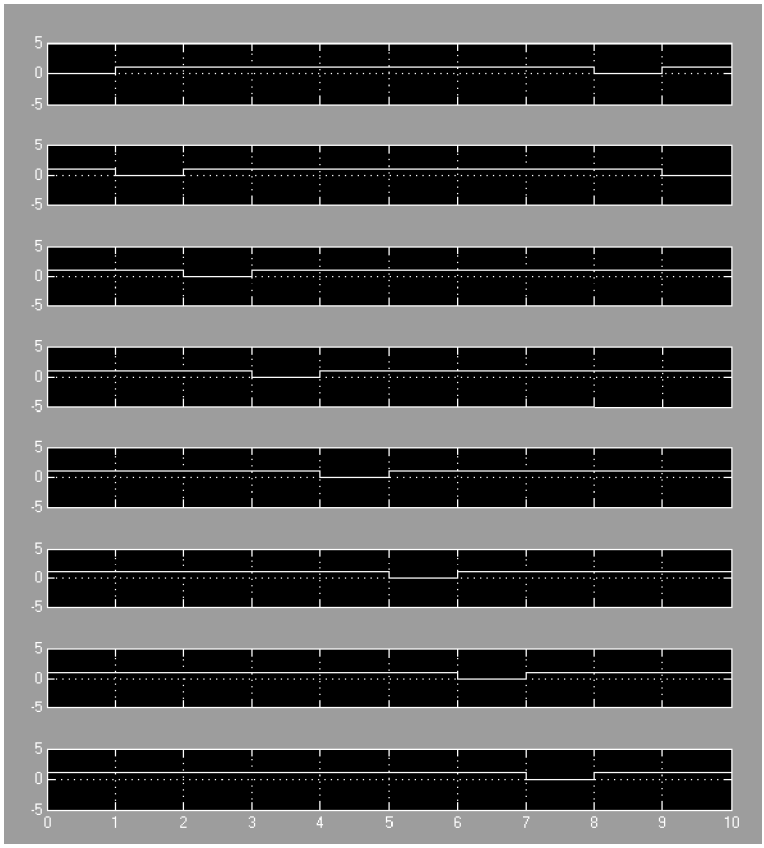


图 6-3-8　数字 1～8 所对应的波形

图 6-3-9 是编码器的输出波形，可以看出，输出的三位二进制码（Y2Y1Y0）依次是 000、001、010、011、100、101、110、111，实现了编码的功能。

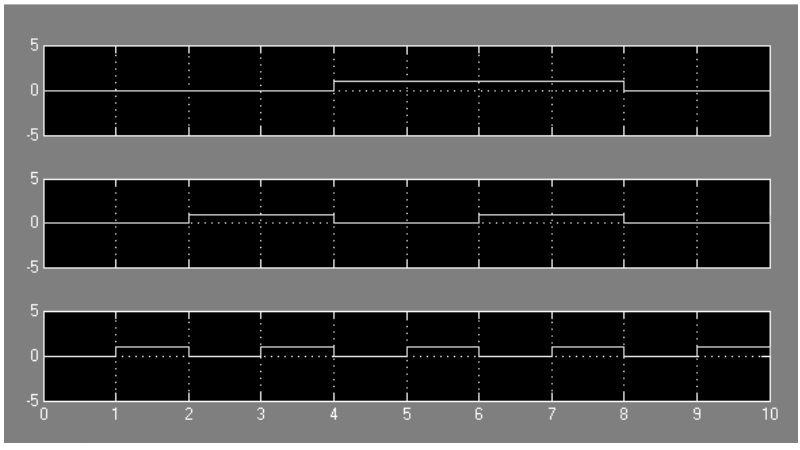


图 6-3-9　数字 1～8 所对应的二进制波形

2．弹跳的皮球

假设在 10 m 高度以初速度 20 m/s 向上抛出一皮球，试仿真皮球的运行速度及位移曲线。

仿真分析：由力学定理可得到如下速度及高度的方程组，假设小球落地后能量损失 80%。

$$\begin{cases} v(t) = 20 + \int_0^t g\mathrm{d}t, & g = -9.81 \\ h(t) = 10 + \int_0^t v(t)\mathrm{d}t \times 0.5 \\ \text{when } h = 0, & v \Rightarrow -0.8v \end{cases}$$

系统仿真模型见图 6-3-10。

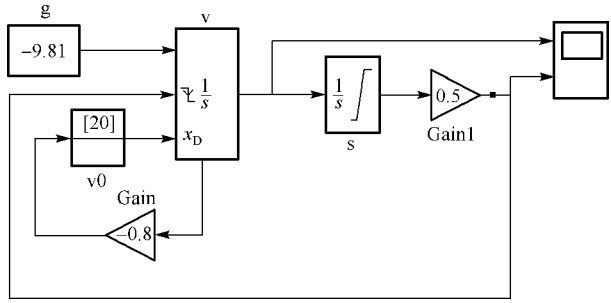


图 6-3-10　弹跳的皮球仿真模型

其中速度积分器的设置如图 6-3-11 所示，初始参数由外部常量提供，速度为 20 m/s，勾选“过零检测”。

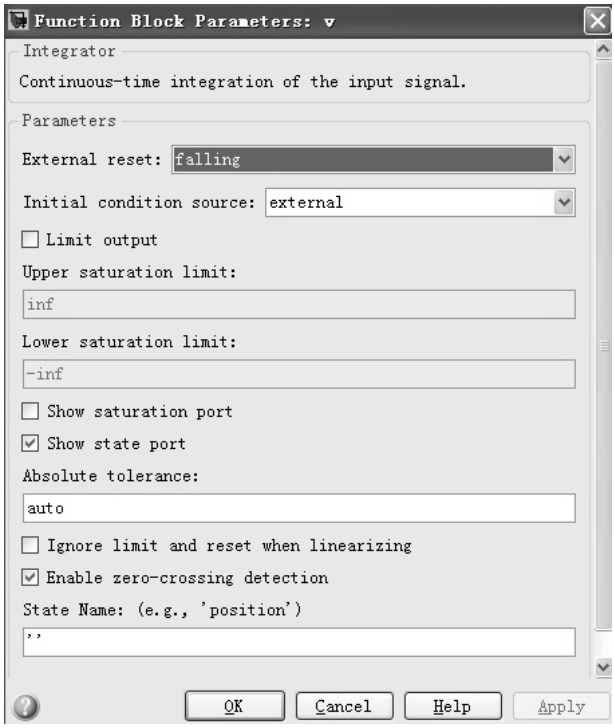


图 6-3-11　速度积分器参数设置

位移积分器的设置如图 6-3-12 所示，初始位移为 10。

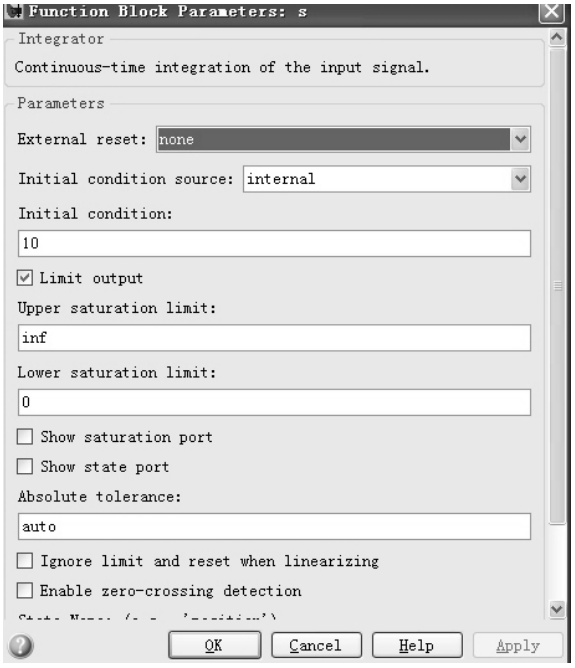


图 6-3-12　位移积分器参数设置

图 6-3-13 给出了小球弹跳时速度及位移的近似走势图。

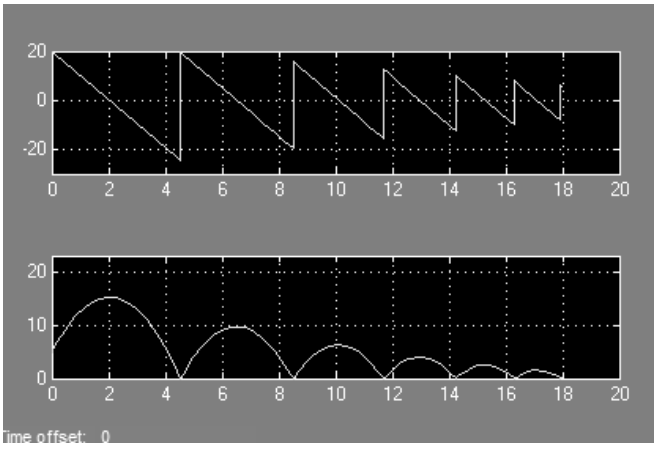


图 6-3-13　速度和位移的近似走势

6.4　子系统

当模型的规模较大时，可以把几个模块组合成一个新的模块，这样的模块称为子系统。子系统是系统构成的一部分，表现形式为具有几个输入/输出端口的模块，内部结构在系统中不表现出来。建立子系统有助于简化系统结构，提高系统设计的层次性和重用性。子系统相关模块均在 Port & subsystems 库中。

6.4.1 子系统的建立及封装

1．子系统的建立

子系统建立方法有两种：

- 直接选中已有模块，形成子系统；
- 添加一个 Subsystem 模块到模型中，然后进行编辑。

Simulink 用 Inport 模块和 Outport 模块代表该子系统外部的输入和输出。

2．子系统的封装

对于一个已经设计成熟的子系统，将其内部结构隐含起来，以便访问该模块时只出现参数设置对话框，模块中所需要的参数可以由这个对话框来输入，可以将其各项性能标准化以供其他用户使用，这就是封装功能。封装后的子系统与 Simulink 提供的模块一样拥有图标，并且用鼠标左键双击图标时会出现一个用户自定义的参数设置对话框，实现在对话框中设置子系统中的参数。使用封装的优点是：

- 可以向子系统模块中传递参数；
- 隐藏子系统中不需要过多展现的内容；
- 保护子系统中的内容，防止模块被随意篡改。

封装后，选中子系统图表，执行 edit/edit mask，显示封装设置窗口如图 6-4-1 所示，在该窗口中可以设置封装参数。

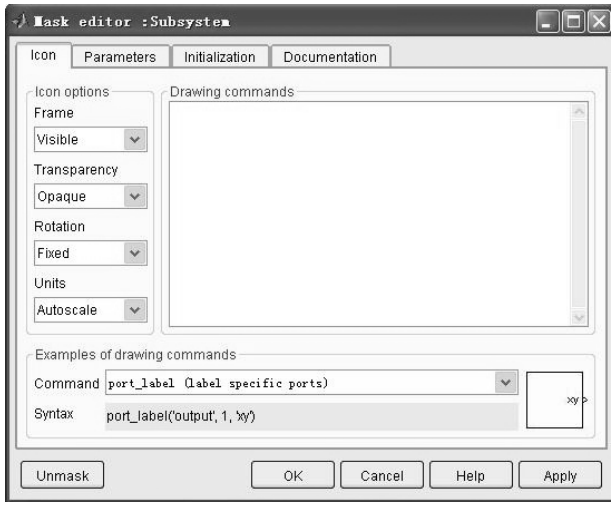


图 6-4-1　模块封装对话框（icon 页）

（1）图标编辑对话框：图标显示界面控制参数。

- icon frame：设置图标边框为可见或不可见；
- icon transparency：设置图标为透明或不透明；
- icon rotation：设置图标为固定或可旋转显示；
- drawing coordinates：设置图标绘制命令所使用的坐标系单位；

- command：图标绘制命令栏。

（2）参数设置对话框：设置参数名称及描述等信息。这些参数即为子系统封装后用于与上层系统交互传递的参数。

（3）初始化设置对话框：initialization commands（初始化命令栏）。一般为 MATLAB 命令，在此可定义封装后子系统工作空间中的各种变量，这些变量可以被封装子系统模块图标绘制命令、其他初始化命令或子系统中的模块使用。

（4）文档对话框：设置封装类型，封装描述，帮助文档。

3．PID 控制子系统的设计

PID 控制是控制系统中运用最多的一种控制方式，PID 控制即比例、积分、微分控制，可通过设置比例、积分、微分的参数建立通用子系统，供其他系统调用。PID 控制的数学模型为

$$U = K_{\mathrm{p}}\left(e + \frac{1}{T_{\mathrm{i}}}\int e + \frac{1}{T_{\mathrm{d}}}\frac{\mathrm{d}e}{\mathrm{d}t}\right)$$

子系统模型如图 6-4-2 所示，其中用到的模块有微分模块、积分模块、增益模块、数学相加模块等。

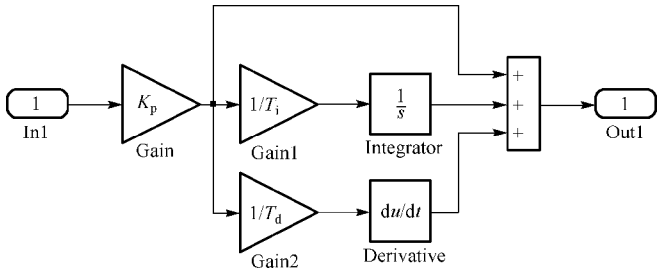


图 6-4-2　子系统模型

PID 控制子系统建立后，并对其进行封装，设定相应的 P、I、D 三个参数及模型描述，如图 6-4-3 所示。

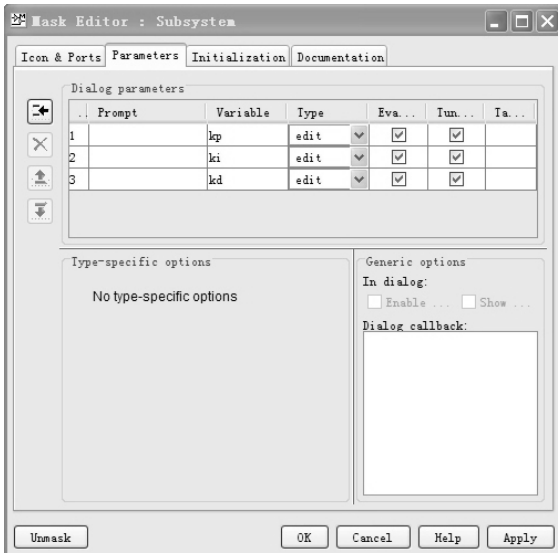


图 6-4-3　PID 子系统参数

保存模型，使用时，双击该模块，输入三个数值即可实现比例、积分、微分三个参数设置。也可以把建立的 PID 控制模块填加到系统模型库中，供以后使用。

6.4.2 条件子系统

子系统的执行可以由输入信号（控制信号）来控制，此类子系统称为条件子系统。条件子系统包括使能子系统、触发子系统、使能触发子系统等。

1．使能子系统（Enable Subsystems）

当子系统由输入信号控制时，输入信号由负变正时子系统开始执行，直到输入信号再次变为负时结束。输入信号可以是标量或向量。若为向量有一个信号元素大于 0 就执行。

【例 6-4-1】利用使能子系统构成一个正弦半波整流器。

正弦半波整流器模型如图 6-4-4 所示。

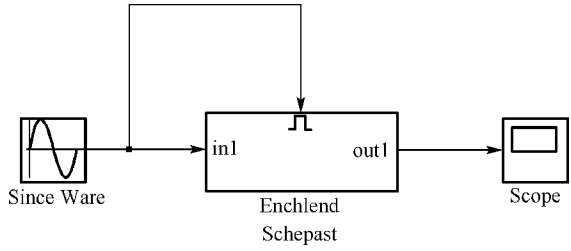


图 6-4-4　正弦半波整流器模型

对 Enable 设置参数，选中 Show output port 可以增加一个输出端，注意此时子系统内外均需连接好，如图 6-4-5 所示。

正弦半波整流器结果波形如图 6-4-6 所示。

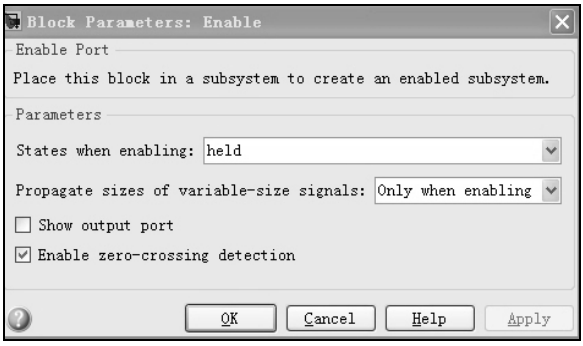


图 6-4-5　使能条件设置

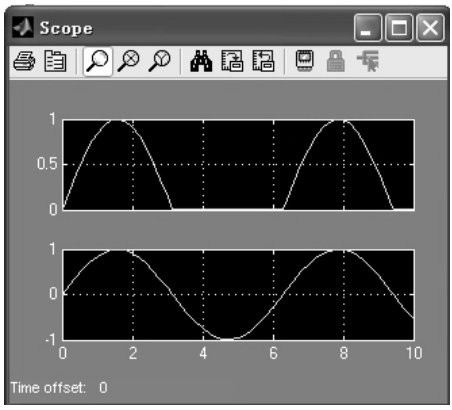


图 6-4-6　正弦半波整流器结果波形

2．触发子系统（Trigger Subsystems）

当触发事件发生时开始执行子系统。每次触发结束到下次触发之前总是保持上一次的输出值，而不会重新设置初始值。触发事件有 4 种类型，即上升沿触发、下降沿触发、跳

变触发和回调函数触发。双击触发子系统中的触发器模块（Trigger），在弹出的对话框中可选择触发类型。

【例 6-4-2】设计一个触发子系统，要求触发器为下降沿触发，正弦输入经触发控制后，成为阶梯波。

触发子系统仿真模型见图 6-4-7。

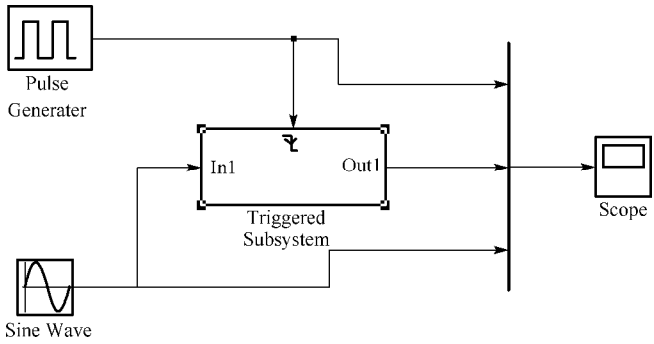


图 6-4-7　触发子系统模型

系统仅在脉冲信号的下降沿导通，并保持导通时刻的输入值至下一个脉冲下降沿，如图 6-4-8 所示。

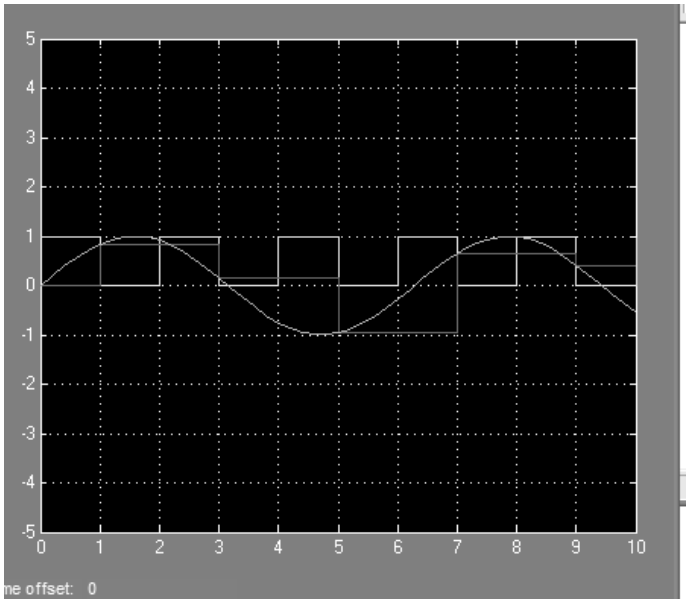


图 6-4-8　触发子系统结果波形

3．使能触发子系统（Enabled and Triggered Subsystems）

使能和触发条件共同作用于子系统执行，即只用于信号正时触发事件发生。端口和模块库中还有很多控制系统流程的子系统，如 if、while、switch，功能类似于编程中的编程控制，在此不一一叙述。

6.5 S-函数

6.5.1 S-函数的功能

S-函数是系统函数（System Function）的简称，是扩展 Simulink 功能的强有力工具，当有些过程用普通的 Simulink 模块不容易搭建时，用户可以利用 MATLAB、C 语言、C++语言等程序创建自己定义的 Simulink 模块。例如，MATLAB 语言编写的 S-函数可以充分利用 MATLAB 所提供的丰富资源，方便地调用各种工具箱函数和图形函数；使用 C 语言编写的 S-函数则可以实现对操作系统的访问，如实现与其他进程的通信和同步等。

非 MATLAB 语言编写的 S-函数需要编译为 Mex 文件，Simulink 可以随时动态地调用这些文件，构成 S-函数模块，像标准 Simulink 模块那样直接调用。

实际上 Simulink 许多模块所包含的算法均是用 S-函数写的，用户也可以编写自己的 S-函数，然后进行封装便可得到具有特定功能的定制模块。S-函数使 Simulink 更加充实、完备，具有更强的处理能力。此外，S-函数在带有代数环的模型中还可以改善仿真的效率。

S-函数使用一种特殊的调用规则来使得用户可以与 Simulink 的内部解法器进行交互，这种交互使得 Simulink 内部解法器与内置的模块之间的交互非常相似，而且可以适用于不同性质的系统，如连续系统、离散系统以及混合系统。

6.5.2 S-函数的调用

在 Simulink 使用 S-Functions 的方法就是从 Simulink 中的 User-Defined Functions 模块库中向 Simulink 模型文件窗口中拖放 S-Function 模块，然后在 S-Functions 模块的对话框中 S-Functions Name 框中输入 S-函数的文件名，在 S-Functions Parameters 框中输入 S-函数的参数值。

单击 Edit 的选项可以编辑 S-函数的代码部分，系统提供了一个模板供代码部分的修改。

S-函数的使用比一般函数复杂，由一种特定的语法构成，用来描述并实现连续系统、离散系统以及复合系统等动态系统；S-函数能够接收来自 Simulink 求解器的相关信息，并对求解器发出的命令做出适当的响应，这种交互作用类似于 Simulink 系统模块与求解器的交互作用。一个结构体系完整的 S-函数包含了描述动态系统所需的全部能力，所有其他的使用情况都是这个结构体系的特例。往往 S-函数模块是整个 Simulink 动态系统的核心。

6.5.3 S-函数的编写规则

MATLAB 提供了一个模板文件 sfuntmpl.m，该模板文件位于 MATLAB 根目录 toolbox/Simulink/blocks 下。在该模板中包括了定义函数所必须的程序代码，同时还附加了详细的注释。建议用户在编写自己的 S-函数时，先将该文件拷贝成自己的文件，然后对其内容进行修改，以尽可能减少因录入而造成的错误。

S-函数主程序的引导语句为：

```
function[sys,x0,str,ts]=fname(t,x,u,flag)]
```

其中默认的 4 个输入参数 t、x、u 和 flag（次序不能变动），各自代表的意义是

- t：表示当前仿真时刻，是采用绝对计量的时间值，从仿真开始模型运行时间的计量值；
- x：模块的状态向量，包括连续状态向量和离散状态向量；
- u：模块的输入向量；
- flag：执行不同操作的标记变量。

应注意的是，S-函数中有关 flag 的条件判断应与系统结构的定义相对应。例如，若未定义连续状态变量，则 S-函数中不应出现 flag=1 的判断。系统结构用一向量定义，该向量的六个分量依次表示连续状态向量的个数、离散状态向量的个数、输出量的个数、输入量的个数、系统根的个数、系统是否有直馈（即输出直接依赖输入）。

默认的 4 个返回参数为 sys、x0、str 和 ts，它们的次序也不能改变，代表的意义为

- sys：通用返回函数；
- x0：初始状态值，当 flag 的值为 0 时才有效；
- str：没有明确定义，为将来应用作保留；
- ts：一个 m×2 矩阵，它的两列分别表示采样时间间隔和偏移。

模板文件中可用的子函数说明如下。

- mdlInitializeSizes：初始化（设置各种参数值），包括采样时间、连续或者离散状态的初始条件和 sizes 数组；
- mdlDerivatives：计算连续状态变量的微分方程，输出值 sys 为状态值的微分；
- mdlUpdate：更新离散状态、采样时间和主时间同步的要求，输出值 sys 为状态值在下一时刻的更新值；
- mdlOutputs：计算 S-Function 的输出，输出值 sys 为输入值与状态值的函数；
- mdlGetTimeOfNextVarHit: 计算下一个采样时间点的绝对时间；
- mdlTerminate：结束仿真任务。

在上述六个子程序中，mdlInitializeSizes，mdlDerivatives(t,x,u)和 mdlOutputs(t,x,u)是三个最基本的也是经常使用的子程序，若熟练掌握它们的用法，将会大大简化仿真过程。

6.5.4 S-函数实例

【例 6-5-1】采用 S-函数构造非线性分段函数。

$$y=\begin{cases}3\sqrt{x} & x<1\\ 3 & 1\leqslant x<3\\ 3-(x-3)^2 & 3\leqslant x<4\\ 2 & 4\leqslant x<5\\ 2-(x-5)^2 & 5\leqslant x<6\\ 1 & x\geqslant 6\end{cases}$$

操作步骤如下：

（1）根据数学模型，编写 S-函数。

```
function [sys,x0,str,ts]=sfunction(t,x,u,flag)
switch flag,
case 0,
   [sys,x0,str,ts]=mdlInitializeSizes;
case 3,
   sys=mdlOutputs(t,x,u);
case {1,2,4,9}
   sys=[];
otherwise
   error(['Unhandled flag=',num2str(flag)]);
end
function[sys,x0,str,ts]=mdlInitializeSizes
sizes= simsizes;
sizes.NumContStates= 0;
sizes.NumDiscStates= 0;
sizes.NumOutputs= 1;
sizes.NumInputs= 1;
sizes.DirFeedthrough= 1;
sizes.NumSampleTimes= 1;
sys=simsizes(sizes);
x0=[];
str=[];
ts=[0 0];
function sys=mdlOutputs(t,x,u)
if u<1
   sys=3*sqrt(u);
elseif u>=1&u<3
   sys=3;
elseif u>=3&u<4
   sys=3-(u-3)^2;
elseif u>=4&u<5
   sys=2;
elseif u>=5&u<6
   sys=2-(u-5)^2;
else
   sys=1;
end
```

（2）完成 S-函数的编写后，接着建立 Simulink 模型，将功能模块 S-function 复制到设计区域，打开其参数页，输入 S-函数的文件名 sfunction.m。S-函数使用如图 6-5-1 所示。

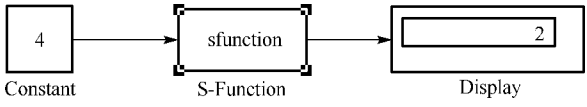


图 6-5-1　S-函数使用

（3）运行 Simulink 模型，可查看计算结果。

6.6 性能优化

6.6.1 仿真性能和精度

在 Simulink 中，仿真的性能和精度受许多因素的影响，包括模型设计和仿真参数选择，建模时应注意以下事项。

（1）建模操作。建模时应该把要解决的问题思考清楚，首先在纸上画出草图，然后在计算机上输入，所有模块都复制到模型窗口里以后，再链接，采用这些方法，将有助于减少打开文件所需要的时间，提高工作效率。

（2）模型的整理。模型的结构要符合一般制图的规范，清晰，整齐有助于理解阅读。给模块加的注释名称要说明模块的主要用途，以便于分析整理。

（3）少用 MATLAB 的 Fcn 模块。MATLAB 的 Fcn 模块每进行一步就要调用解释程序，会大大降低仿真速度，应尽量少用。

一般情况下，仿真器能在默认参数值时精确、有效地处理大多数的模型仿真。然而，如果适当调整模型的仿真器和仿真参数，会产生更好的结果。也就是说，如果知道该模型的性能信息并提供给仿真器，仿真的结果也会更好。

1．加快仿真速度

有许多因素都会引起仿真速度减慢。

（1）模型包含 MATLAB Fcn 模块，尽可能地使用内置 Fcn 模块。

（2）模型中包含 M 文件的 S-函数。M 文件的 S-函数也在每一步引起 MATLAB 解释器的调用。可以考虑将 S-函数转换为一个子系统或者一个 C-MEX 的 S-函数。

（3）模型包含 Memory 模块。使用 Memory 模块会使变阶仿真器（odel5s 和 ode1l3）在每一步将阶数重复设置为 l。

（4）最大步长太小。如果用户改变了最大步长，必须使用默认值（auto）再一次运行该仿真。

（5）精度要求太高。默认的相对容限（精度 0.1%）通常是足够的，对于有零状态的模型，如果绝对容限参数太小，接近零状态值会需要太多的步。

（6）时间刻度太大，可以减小时间间隔。

（7）使用非刚性仿真器去解决刚性问题，可尝试使用 odel5s 仿真器。

（8）模型中使用的采样时间彼此间不成倍数关系。这样的混合采样时间会使仿真器采用过小的步长，以保证采样时间能精确地反映所有的采样时间，导致运行速度降低。

（9）模型中包含代数环。代数环的解法是在每步进行迭代运算，因此大大降低了执行性能。

2．提高仿真精度

仿真精度可影响系统的稳定性和精确性，可以从以下几方面考虑。

（1）检查仿真精度。在一个合理的时间段内运行仿真，然后将相对误差容限（Relative

To lerance）减少到 1e-4（默认为 1e-3）或减小绝对误差容差，并再次运行仿真。比较两者的仿真结果，如果结果没有大的不同，可以确信该算法收敛。

（2）如果仿真在开始时就漏过了关键步的操作，则必须减少初始步长（Initial Step Size），确保仿真不“越过”这些关键操作。

（3）如果仿真结果在时间上不稳定，则意味着：

① 用户系统可能是不稳定的。

② 如果用户使用的是 odel5s，则需要限制最大阶为 2，或者尝试使用 ode23s 仿真器。

（4）如果仿真结果不精确，则意味着：

① 对于含有接近零值状态的模型，如果绝对误差容限参数太大。仿真会在零值状态周围采用过少的操作步。减小该参数值或者在 Integrator 模块对话框内调整它的各自状态。

② 如果减少绝对误差容限并未有效地提高精度，则减少相对误差容限参数，来减小可接受误差和强迫使用小步长和多步操作。

6.6.2 代数环

1．代数环的概念

在仿真中，当输入信号直接取决于输出信号，同时输出信号也直接取决于输入信号时，由于数字计算的时序性，而出现的由于没有输入无法计算输出，没有输出也无法得到输入的“死锁环”，称之为代数环。

Simulink 的系统模块中，有些模块输入端口具有直接馈通的特性。所谓模块的直接馈通，是指如果在这些模块的输入端口中没有输入信号，则无法计算此模块的输出信号。在 Simulink 中具有直接馈通特性的模块有以下的几种。

（1）Math Function 数学函数模块。

（2）Gain 增益模块。

（3）Product 乘法模块。

（4）State-Space 状态空间模块（其中矩阵 ***D*** 不为 0）。

（5）Transfer Fcn 传递函数模块（分子与分母多项式阶次相同）。

（6）Sum 求和模块。

（7）Zero-Pole 零极点模块（零点与极点数目相同）。

（8）Integrator 积分模块。

计算机在处理代数环计算时，采用的是一种迭代算法，也就是说在每个时间步里，计算机都将进行这样的计算，这会使得计算机的计算时间增长。同时，由于这种代数环包含的模块的“无延时”特性，要求环上所有模块的输出在同一时刻计算，这与系统顺序仿真的要求不符。代数环的出现将会严重降低系统的仿真速度甚至于降低仿真精确度或得到错误的仿真结果。

系统模型中产生代数环的条件如下：

（1）具有直接馈通特性的系统模块的输入，直接由此模块的输出来驱动。

（2）具有直接馈通特性的系统模块的输入，由其他直接馈通模块所构成的反馈回路间接来驱动。

图 6-6-1 所示为一个非常简单的标量代数环的构成。

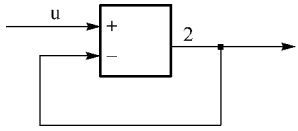


图 6-6-1　一个非常简单的标量代数环结构

从数学角度来讲，该循环暗示 $z=u-z$，即 $z=\frac{1}{2}u$，这就是求解表达式，但大多数代数循环就无法通过这种检查自动计算。

2．解决方案

要解决代数环问题，可以采取以下方法。

（1）设计模型时尽量不采用代数环设计；

（2）在计算速度可以忍受的范围内，可以不必介意代数环问题；

（3）对代数环采取代数约束；

（4）切断模型中的代数环。

- 引入小时间延迟，使信号不同时发生；
- 引入 Memory 模块，使返回上一个采样周期的值；
- 引入滤波器。

Algebraic Constraint模块是建模代数方程和指定初始化估值的一个方便的方法。该模块将它的输入信号 *f*(*z*)限制到 0，并输出代数状态 *z*。该模块输出一个值，该值对于在模块的输入端产生一个零点是必须的。输出必须通过反馈直接影响到输入，例如，该反馈通道单独包含带有直接注入的模块。在代数限制模块对话框中，可以提供一个初始的代数状态值以改善代数循环求解器的效率。

为了在仿真或更新时使 Simulink 将自己探测到的代数循环设定为致命错误，可在 Configuration Parameters 对话框的 Diagnostics 面板上，设置 Algebraic loop 为 error。

3．消除代数循环

【例 6-6-1】求方程组 $x=u-0.5y,\ y=4x+2\dot{x}$ 的解。

按照原始方程所建的模型如图 6-6-2 所示。

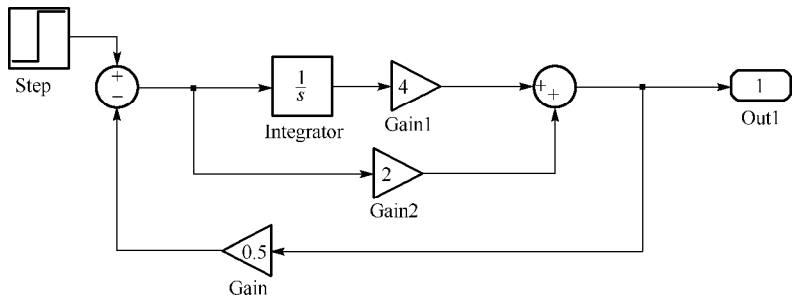


图 6-6-2　按原始方程所建的模型

观察模型，包含增益直通模块形成代数环，通过重组模型，直接消除代数环，建立等价模型，如图 6-6-3 所示。

也可通过加入 Memory 模块，切断代数环，如图 6-6-4 所示。

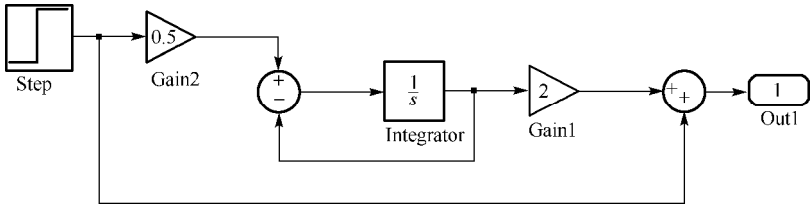


图 6-6-3　消除了代数环的等价模型

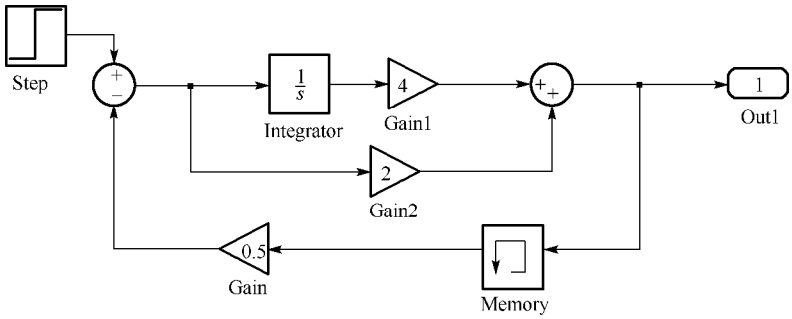


图 6-6-4　用 memory 模块切断代数环的模型

6.6.3　过零检测

1．过零的产生

在动态系统的仿真过程中，所谓过零，是指系统模型中的信号或系统模块特征的某种改变。这种特征改变包括以下两种情况。

（1）信号在上一个仿真时间步长之内改变了符号。

（2）系统模块在上一个仿真时间步长改变了模式（如积分器进入了饱和区段）。

在动态系统中，状态变量的不连续性往往表征了系统的重要事件。以篮球撞击地面系统为例，在篮球撞击地面的时刻，撞击的位置是不连续的，在仿真中，如果篮球是在两次时间步之间撞击了地面，那么仿真的结果可能是篮球在半空中就已经翻转方向，这与实际情况是不相符的，因此，对不连续点的精确仿真是非常重要的。

【例 6-6-2】绘制正弦函数绝对值。

程序如下：

```
t=0:0.2:2*pi;
y=abs(sin(t.^2));
plot(t,y)
```

例 6-6-2 的结果如图 6-6-5 所示，观察图形，出现过零点未检测问题。

过零检测可以使 Simulink 精确地仿真不连续点，而不必过多地选用小步长。每个采样点仿真结束时 Simulink 检测是否有过零函数符号变化，如果检测到过零点，Simulink 将在前一个采样点和目前采样点间内插值。

事实上，Simulink 中的许多模块都支持过零检测。在实际仿真中，如果用户对所有的

系统，包括含有不连续状态的系统，利用过零检测进行仿真，不仅可以加快仿真速度，也可以提高仿真精度。

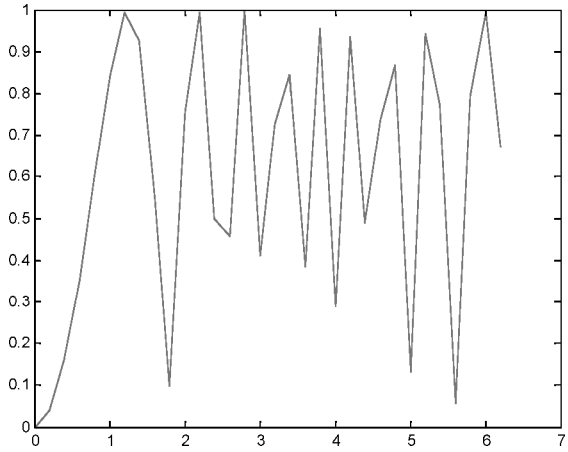


图 6-6-5　没有过零检测的函数图形

对于 Simulink 中不同模块来说，产生过零的类型是不同的。一般而言，系统模型中模块过零的作用有两种类型：一是用来通知求解器，系统的运行模式是否发生了改变，也就是系统的动态特性是否发生改变；二是驱动系统模型中其他模块。过零信号包含三种类型：上升沿、下降沿、双边沿。

2．事件通知

在动态系统仿真中，采用变步长求解器可以使 Simulink 正确地检测到系统模块与信号中过零事件的发生。当一个模块通过 Simulink 仿真环境通知求解器，在系统前一仿真步长时间内发生了过零事件，变步长求解器就会缩小仿真步长，即使求解误差满足绝对误差和相对误差的上限要求。缩小仿真步长的目的是判定事件发生的准确时间（也就是过零事件发生的准确时刻）。

3．解决方案

（1）缩小步长。

（2）仿真时，减小 RelTol 选项。

（3）选择过零点检测。

在 Simulink 的模块库中，并非所有的模块都能够产生过零事件。

对于不具有过零检测的能力的模块而言，可以使用信号与系统库（Signals & Systems）中的 Hit Crossing 零交叉模块来实现过零检测。当 Hit Crossing 模块的输入穿过某一偏移值（offset）时会产生一个过零事件，可以用来为不带过零检测能力的模块提供过零检测。

第 7 章　控制系统的仿真

控制系统仿真主要是控制系统的分析和设计，都是以数学模型为基础的，也是工程中解决问题的主要方法。MATLAB 已成为控制系统设计的标准化平台，它提供的针对控制领域丰富的工具集和强大的计算能力，使得控制设计人员无论是在进行数据分析、算法设计，还是系统仿真、产品实现等方面都得心应手。

7.1　控制系统的基础理论

7.1.1　控制系统的组成

控制理论最基本的任务是对给定的被控系统，设计能满足所期望的性能指标的闭环控制系统，即寻找反馈控制律。

一个典型的闭环控制系统中相应信号描述如图 7-1-1 所示。

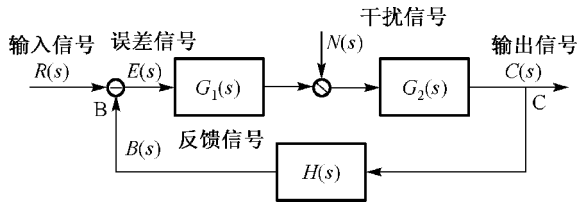


图 7-1-1　典型的闭环控制系统结构图

图中各信号和装置的含义如下：

- 输入信号 $R(s)$：系统的外部输入量，即系统给定值。
- 输出信号 $C(s)$：系统的输出量，即系统被控量。
- 反馈信号 $B(s)$：主反馈环节的输出信号。
- 误差信号 $E(s)$：输入信号与反馈信号之差 e=r–b。
- 干扰信号 $N(s)$：内部和外部的干扰量。
- 控制器 $G_1(s)$：系统中承担信号放大、传动和执行作用的装置。
- 被控对象 $G_2(s)$：系统中的控制对象。
- 反馈环节 $H(s)$：用于检测输出状况的测量装置。
- 前向通道：从系统输入端到输出端的正向传输通道，且每个节点只通过一次。
- 反馈通道：从输出端 C 反馈到输入端 B 的传输通道。
- 反馈回路：信号从前向通道与反馈通道连续传输的闭合回路。
- 比较环节：在系统中进行信号叠加的作用点，以产生偏差信号。

通常，研究系统运动规律的问题称为分析问题；研究改变运动规律的可能性和方法的问题则为综合（或设计）问题。

7.1.2 控制系统的分类

根据实际工程应用，控制系统有多种分类方法。

1．按控制系统是否形成闭合回路分类

1）开环控制系统

一个控制系统，如果在其控制器的输入信号中不包含受控对象输出端的被控量的反馈信号，则为开环控制系统，开环系统的特点是容易受各种干扰的影响，控制精度较低，但结构简单、成本低，也容易实现，所以可用在对控制要求不高的小型机器设备上。

2）闭环控制系统

一个控制系统，如果在其控制器的输入信号中包含来自受控对象输出端的被控量的反馈信号，则为闭环控制系统，或为反馈控制系统，这是闭环控制最常见的控制方式。

2．按信号的结构特点分类

1）反馈控制系统

反馈控制系统是根据被控量和给定值的偏差进行调节的，最后使系统消除偏差，达到被控量等于给定值的目的。

2）前馈控制系统

前馈控制系统直接根据扰动信号进行调节，扰动量是控制的依据，由于它没有被控制量的反馈信号，故不形成闭合回路，所以它是一种开环控制系统。

3）前馈-反馈复合控制系统

它是在反馈控制系统的基础上增加了对主要扰动的前馈补偿作用。

3．按给定值信号的特点分类

1）恒值控制系统

恒值控制系统中控制系统的任务是保持被控量恒定不变，这是生产过程中用得最多的一种控制系统。

2）随动控制系统

随动控制系统是给定信号随时间的变化规律事先不能确定的控制系统，其任务是在各种情况下快速、准确地使被控量跟踪给定值的变化。

3）程序控制系统

它的给定值按事先预定的规律变化，是一个已知的时间函数，控制的目的是要求被控量按确定的给定值的时间函数来改变。

4．按控制系统元件的特性分类

1）线性控制系统

当控制系统的各元件的输入/输出特性是线性特性，控制系统的动态过程可以用线性微分方程来描述，称这种系统为线性控制系统。其特点是可以应用叠加原理。

2）非线性控制系统

当控制系统中有一个或一个以上的非线性元件时，系统的特性就要用非线性方程来描述，由非线性方程描述的控制系统为非线性控制系统。

5．按控制系统信号的形式分类

1）连续控制系统

当控制系统的传递信号都是时间的连续函数，这种系统称为连续控制系统。

2）离散控制系统

控制系统在某处或几处传递的信号是脉冲系列或数字形式，在时间上是离散的，这种系统称为离散控制系统。

7.1.3 控制系统的数学模型

数学模型是描述系统（或元件）的动态特性的数学表达式，它是分析和设计控制系统的基础。

在时间域进行分析的时候，控制系统的数学模型主要为描述输入和输出关系的微分方程或者差分方程；在复频域进行分析的时候，主要用 S 传递函数或者 Z 传递函数来描述系统输入和输出关系。用微分方程（或差分方程）和传递函数描述的模型形式也称为输入/输出模式，它只描述系统的输出变量，对系统内部的其他变量不给出任何信息。因此有时候输入/输出模式对系统的描述是不完全的。

如果要知道系统内部的其他变量，则必须确定一组独立的状态变量，用一组一阶的微分方程或者差分方程来描述系统内部状态的变化和转移情况，再用一个由这一组状态变量组合后得到的输出方程来描述系统的输出，这样由状态方程和输出方程组成的数学模型在现代控制理论中称为状态变量模式。

系统按性能有很多种，如线性和非线性、定常和时变、连续和离散、确定和不确定，不同的系统采用不同的方式建模，其中线性定常系统是最基础也是研究得比较透彻的系统。

1．微分方程

描述系统动态特性的微分方程可以通过具体系统的物理定律，如机械系统中的牛顿定律，电气系统中的基尔霍夫定律等来获得。

对单输入-单输出系统，其动态特性可用下列微分方程式表示为

$$a_n\frac{\mathrm{d}^n x_\mathrm{o}(t)}{\mathrm{d}t^n}+a_{n-1}\frac{\mathrm{d}^{n-1}x_\mathrm{o}(t)}{\mathrm{d}t^{n-1}}+\cdots+a_1\frac{\mathrm{d}x_\mathrm{o}(t)}{\mathrm{d}t}+a_0x_\mathrm{o}(t)$$
$$=b_m\frac{\mathrm{d}^m x_\mathrm{i}(t)}{\mathrm{d}t^m}+b_{m-1}\frac{\mathrm{d}^{m-1}x_\mathrm{i}(t)}{\mathrm{d}t^{m-1}}+\cdots+b_1\frac{\mathrm{d}x_\mathrm{i}(t)}{\mathrm{d}t}+b_0x_\mathrm{i}(t)$$

式中，$x_\mathrm{o}(t)$是系统的输出量，$x_\mathrm{i}(t)$是系统的输入量。

如果上述微分方程的系数 a_i（i=0，1，…，n），b_j（j=0，1，…，m）均为常数，则为线性定常微分方程，简称常微分方程，相应的动态系统称为线性定常系统，大多数物理系统均属于这一类，也是研究的重点。

如果上述微分方程的系数中存在一个或一个以上的系数是时间 $x_\mathrm{i}(t)$的函数或是自变量，这类微分方程式称为线性变系数微分方程，相应的系统称为线性时变系统。例如，宇宙飞船控制系统便是一个时变系统，因为随着宇宙飞船上燃料的消耗，飞船质量发生变化，而且当飞船远离地球后，重力也在发生变化。

如果上述微分方程式的系数中，存在一个或者一个以上的系数是因变量 $x_\mathrm{o}(t)$的函数，那么此微分方程称为非线性微分方程，相应的系统也称为非线性系统。例如，电气系统中某些元件存在继电特性、饱和、死区和磁滞等现象，均使这种系统成为非线性系统。

2．传递函数

微分方程是在时域中描述系统动态性能的数学模型，传递函数是在复频域中描述系统动态性能的数学模型。对 SISO 系统的微分方程在零初始条件下做拉氏变换，则可得系统的传递函数，即

$$G(s)=\frac{y(s)}{u(s)}=\frac{b_1s^m+b_2s^{m-1}+\cdots+b_{m+1}}{a_1s^n+a_2s^{n-1}+\cdots+a_{n+1}}$$

传递函数不仅可以表征系统的动态特性，而且可以研究系统的结构或参数变化对系统性能的影响。传递函数是经典控制理论中最基本也是最重要的概念。

3．零极点模型

零极点模型是传递函数的另一种表现形式，其原理是分别对原系统传递函数的分子和分母进行分解因式处理，以获得系统的零极点表示形式，对于 SISO 系统：

$$G(s)=\frac{k(s+z_1)(s+z_2)(s+z_3)\cdots(s+z_m)}{(s+p_1)(s+p_2)(s+p_3)\cdots(s+p_n)}$$

式中，z_i 为零点，p_j 为极点，k 为系统增益。

由传递函数的定义可知，传递函数的分母是系统齐次微分方程的特征多项式，所以传递函数的极点是微分方程的特征根，它决定着所描述系统固有自由运动的模态，即系统本身的运动特性。而传递函数的零点在分子上，它将影响到各模态在运动中所占的比重，决定着对所描述系统控制作用的动态效果。对零点和极点的研究，是分析和设计系统的一个重要方法。

4．状态空间描述

状态方程与输出方程的组合称为状态空间表达式，又称为动态方程，经典控制理论用传递函数将输入/输出关系表达出来，而现代控制理论则用状态方程和输出方程来表达输入/输出关系，揭示了系统内部状态对系统性能的影响。

$$\dot{x} = \boldsymbol{A}x + \boldsymbol{B}u$$
$$y = \boldsymbol{C}x + \boldsymbol{D}u$$

式中，y 和 u 为外部变量，x 为内部变量，$\boldsymbol{A}$ 为状态矩阵（系统矩阵），$\boldsymbol{B}$ 为输入矩阵，$\boldsymbol{C}$ 为输出矩阵，$\boldsymbol{D}$ 为前馈矩阵（直接传输矩阵）。

状态空间描述是现代控制理论的基础，它不仅可以描述输入输出关系，而且可以描述系统的内部特性，特别适合于多输入多输出系统，也适用于时变系统、非线性系统和随机控制系统。从这个意义上讲，状态空间描述是对系统的一种完全描述。

5．结构图描述

利用传递函数模型分析设计系统时，经常使用形象的结构图（框图）。结构图是基于传递函数模型研究控制系统广泛使用的工具，是描述系统各组成元部件之间信号传递关系的数学图形，采用结构图，不仅能方便地求取复杂系统的传递函数，而且能形象直观地表明信号在系统或元件中的传递过程。

绘制系统结构图的步骤如下：

（1）列出各环节（元件）的传递函数。

（2）根据各环节之间的信号流向，用图的形式进行连接。

常见的环节连接方式有串联、并联、反馈。

6．离散时间系统

对于离散系统，系统的输入量，输出量及内部状态量均为时间的离散函数，即时间序列。离散时间系统的数学描述主要有

- 差分方程；
- Z 传递函数；
- 权序列；
- 离散状态空间模型。

对于离散时间系统，对微分方程两边取 z 变换，并设 y, u 及其各阶差分的初始值均为零，可得

$$G(z) = \frac{Y(z)}{U(z)} = \frac{f_m z_m + f_{m-1} z^{m-1} + \cdots + f_1 z + f_0}{g_n z^n + g_{n-1} z^{n-1} + \cdots + g_1 z + g_0}$$

式中，$G(z)$称为系统的 z 函数。

将连续系统离散化进行数字仿真具有以下特点：

（1）将连续系统离散化后进行仿真，可以用得到的离散方程递推求解状态和输出，避免了数值积分方法中繁琐的龙格-库塔系数（导函数）的求取过程，计算方便。

（2）按环节离散化仿真，每步都可求出各环节输入和输出，很容易推广到用于解决非线性系统的仿真问题。

（3）离散化过程中对原连续系统引入了虚拟采样开关和零阶（一阶）保持器，造成的滞后使得仿真计算误差增大，严重时会引起系统数值计算的不稳定。

7．非线性系统

在实际工程中，理想的线性系统是不存在的，绝大多数元器件都具备非线性特性，非线性系统的建模是一项复杂的工作。通常对于典型的非线性模块，仿真工具中能调用已有模块建立其非线性数学模型，大多复杂的非线性系统需进行线性化处理。

所谓线性化，就是在一定的条件下作某种近似，或者缩小一些工作范围，而将非线性微分方程近似地作为线性微分方程来处理。

建立系统线性化数学模型首先要确定系统处于正常工作状态（平衡工作点）时各组成元件的工作点，然后列出各组成元件在工作点附近的增量方程，最后消去中间变量，得到系统以增量表示的线性化微分方程。

增量方程的数学含义就是将参考坐标的原点移到系统或元件的平衡工作点上，对于实际系统就是以正常工作状态为研究系统运动的起始点，这时系统所有的初始条件均为零。

由级数理论可知，若非线性函数在给定区域内存在各阶导数，便可在工作点的邻域将非线性函数展开为泰勒级数。当偏差范围很小时，可略去 2 次以上高次项，从而得到只包含偏差一次项的线性化方程式，实现函数的线性化。这种线性化方法称为小偏差线性化或局部线性化。

小偏差线性化的这种近似，对大多数控制系统来说都是可行的。原因是：

（1）控制系统在通常情况下，都有一个正常、稳定的工作状态，称为平衡工作点。例如，恒温控制系统的正常工作状态是输入、输出为常值（输出为被控温度，输入为期望值）。

（2）当系统的输入或输出相对于正常工作状态发生微小偏差时，系统会立即进行控制调节，力图去消除此偏差，因此可以看出，这种偏差是“小偏差”，不会很大。

几点说明：

① 线性化方程的参数与工作点有关，工作点不同时，得出的参数值也不同。

② 非线性模型的线性化是有条件的，即非线性特性必须是连续的，各阶导数存在，这样才能用泰勒级数展开；变量在工作点附近小范围的变化才能保证其线性化后的模型误差较小。

③ 如果非线性特性是不连续的，则在不连续工作点附近不能得出收敛的泰勒级数，这时就不能对系统进行线性化处理，这类系统称为本质非线性系统。

7.1.4 控制系统的典型环节及传递函数

任何一个复杂控制系统都是由若干典型环节组合而成的，常见的典型环节、传递函数及特点如表 7-1-1 所示。

其中需要注意的是惯性环节与延迟环节的区别，惯性环节从输入开始时刻就已有输出，

仅由于惯性，输出要滞后一段时间才接近所要求的输出值；延迟环节从输入开始后在 0～τ 时间内没有输出，但 $t=\tau$ 之后，输出完全等于输入。

表 7-1-1　控制系统的典型环节及传递函数

典 型 环 节	传 递 函 数	特　　点	实　　例
比例环节	$G(s)=K$	输入/输出量成比例，无失真和时间延迟	电子放大器，齿轮，电位器，感应式变送器等
惯性环节	$G(s)=\dfrac{1}{Ts+1}$	含一个储能元件，对突变的输入，其输出不能立即复现，输出无振荡	RC 网络，一阶水槽等
积分环节	$G(s)=\dfrac{1}{Ts}$	输出量与输入量的积分成正比例，当输入消失，输出具有记忆功能	一阶水槽，电动机角速度与角度间的传递函数等
微分环节	$G(s)=\tau s$	输出量正比输入量变化的速度，能预示输入信号的变化趋势	测速发电机输出电压与输入角度间的传递函数等
振荡环节	$G(s)=\dfrac{1}{T^2s^2+2T\xi s+1}$ $=\dfrac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}$	环节中有两个独立的储能元件，并可进行能量交换，其输出出现振荡	RLC 电路的输出与输入电压间的传递函数等
延迟环节	$G(s)=e^{-\tau s}$	输出量能准确复现输入量，但须延迟一固定的时间间隔	管道压力、流量、皮带运输等物理量的控制，其数学模型就包含有延迟环节

7.1.5　控制系统的性能要求

系统在外加信号的作用下，被控量随时间变化的全过程称为系统的动态过程，也称为过渡过程。系统控制性能的好坏，可以从系统的动态过程中反映出来。

控制系统的性能分析主要是对三大性能——稳定性、稳态性能、动态性能的分析。

在工程实际中，对控制系统的总体性能要求可概括为三个字：稳、准、快，即系统稳定程度高，动态过程平稳性能好；能够较快地到达系统稳态值，暂态响应时间短；最终控制精度高，稳态误差小。

1．动态过程的平稳性

控制系统动态过程的平稳性好，就是要求在系统的平衡值附近，偏移量很小，并且最终趋于稳定值，通常有以下两种动态过程响应。

1）非周期性响应

这种方式不存在振荡过程，系统的响应表现出单调上升的指数规律。

2）周期性响应

这种方式中，系统的输出信号呈现一种周期性的振荡过程。按照振荡过程的特点，可以分为：

① 若输出信号振幅随时间的变化越来越小，最终趋于稳定值，称为衰减振荡；

② 若在平衡值附近输出信号振幅保持不变，称为等幅振荡；

③ 若输出信号的振幅越来越大，称为发散振荡。

输出信号呈现衰减振荡的系统是理想的稳定系统。

2．动态过程的快速性

控制系统动态过程的快速性好，就是要求被控量迅速响应给定信号，偏离平衡值的时间短，这样暂态过程的时间短，可以很快到达稳态。

3．最终响应的精度

控制系统最终响应的精度是指在输入信号作用下，经过暂态到达稳态，系统的输出值与给定值之间的偏差，也称为稳态误差，要得到较高的控制精度应使系统的稳态误差小。

7.1.6 控制系统的分析

在自动控制理论与分析方法中，主要解决以下四个问题，并提供相应的方法和途径。

（1）给出判断系统稳定性的方法，指出稳定性与系统结构、参量之间的关系。

（2）研究系统的控制规律，分析参量与暂态响应的对应关系，提供计算系统暂态响应性能指标的方法。

（3）给出计算稳态误差的方法，指出系统的控制规律、参量与稳态响应之间的关系。

（4）在系统原有状态下，根据需求适当加入校正装置，提高系统的整体性能。

控制系统的分析可在时域和复频域中进行。

1．性能分析

性能分析包括系统的能控、能观性分析、稳定性分析等。

1）能控能观性分析

系统的能控性和能观性是设计控制器和状态观测器的基础，能控性分析是系统输入对状态的控制能力；能观性分析是系统输出对状态的反应能力。

对 n 阶线性定常连续系统

$$\begin{cases} \dot{x} = \boldsymbol{A}x + \boldsymbol{B}u \\ y = \boldsymbol{C}x + \boldsymbol{D}u \end{cases}$$

其能控的充要条件为：能控性矩阵

$$\boldsymbol{Q}_{\mathrm{c}} = (\boldsymbol{B} \quad \boldsymbol{AB} \quad \boldsymbol{A}^2\boldsymbol{B} \quad \cdots \quad \boldsymbol{A}^{n-1}\boldsymbol{B})$$

满秩，即

$$\mathrm{rank}(\boldsymbol{Q}_{\mathrm{c}}) = n$$

其能观的充要条件为：能观性矩阵

$$\boldsymbol{Q}_{\mathrm{o}} = (\boldsymbol{C} \quad \boldsymbol{CA} \quad \boldsymbol{CA}^2 \quad \cdots \quad \boldsymbol{CA}^{n-1})^{\mathrm{T}}$$

满秩，即

$$\mathrm{rank}(\boldsymbol{Q}_{\mathrm{o}}) = n$$

对于离散系统，能控性和能观性有上述类似结论。

2）稳定性分析

在系统特性的研究中，控制系统的稳定性是最重要的指标。如果控制系统稳定，则可以进一步分析系统的其他性能，如果系统不稳定，系统在实际中不能应用。

所谓稳定性是指系统在扰动消失后，由初始偏差状态恢复到原平衡状态的性能。在经典控制理论中，系统稳定的充分必要条件是时间 t 趋于无穷时，系统的单位脉冲响应等于零。

由控制理论的一般规律可知，对线性系统而言：

（1）对于连续时间系统，如果闭环极点全部在 S 平面左半平面，则系统是稳定的。

（2）对于离散时间系统，如果系统全部极点都位于 Z 平面的单位圆内，则系统是稳定的。

2．时域分析

在控制理论中，时域分析是对控制系统进行分析、评价的最直接和最基本的方法。对控制系统进行时域分析，实质上就是研究系统在某一典型输入信号作用下，系统输出随时间变化的曲线，从而分析评价系统的性能。典型输入信号一般采用单位阶跃函数或单位冲激函数。

通常将阶跃响应曲线的几个特征参数作为性能指标，以加单位阶跃干扰，系统衰减振荡过程为例，在单位阶跃输入情况下，对于定值、随动系统典型过程曲线如图 7-1-2 所示。

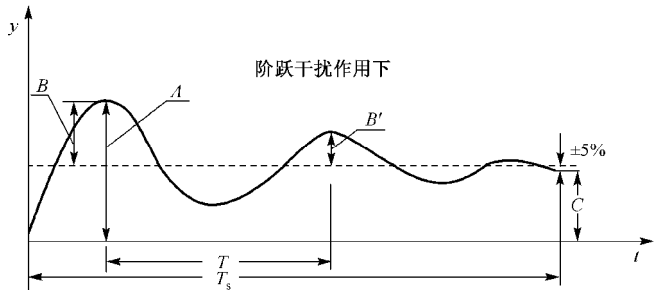


图 7-1-2　过渡过程性能指标示意图

在图 7-1-2 中，各性能指标及在工程中的实际意义如下所述。

1）最大偏差 A

定义：受控变量第一个波峰值与设定值之差。

工程意义：表示偏离设定值的程度（衡量）。偏差越大，时间越长，离生产状态就越远。实际系统中不希望出现，甚至会引起生产事故。

2）余差 C

定义：过渡过程终了时的残余偏差。

工程意义：反映控制准确性的重要指标，生产中一般希望 C 越小越好（不超过某一个预定范围），一般由工艺人员提出受控变量的波动范围所规定。

3）超调量 B

定义：第一个波峰值与稳态值之差。

工程意义：反映过程稳定性的一个指标（同 A）。

4）衰减比 N

定义：第一个波峰与第二个波峰之比。

工程意义：反映过程稳定性的一个指标，$N>1$，衰减振荡；$N=1$，等幅振荡；$N<1$，发散振荡。

5）过渡时间（恢复时间）T_s

定义：从系统受扰后至受控变量又建立新的平衡的最短时间。一般规定为进入稳态值±5%（±2%）范围内所经历的时间。

工程意义：反映过程快慢、长短标志（快速性指标）。在 N 相同的情况下，T_s 越短，过程越快；适应性越强，质量越好。

6）振荡周期 T

定义：从第一个波峰到第二个波峰的时间，周期的倒数为振荡频率 f。

工程意义：在 N 一定的情况下周期短且频率高，反映振荡过程快慢的标志。

3．根轨迹分析

在控制系统分析中，为了避开直接求解高阶多项式的根时遇到的困难，在实践中提出了一种图解求根法，即根轨迹法。控制系统的根轨迹分析方法就是利用系统的某个参数（通常是开环增益）从 0 变化到无穷大时，闭环系统特征根所留下的轨迹（即根轨迹）来分析系统性能以及参数变化对系统性能的影响。

根轨迹可以分析系统参数和结构已定的系统的时域响应特性，以及参数变化对时域响应特性的影响，而且还可以根据对时域响应特性的要求确定可变参数及调整开环系统零极点的位置，并改变它们的个数，根轨迹法可用于解决线性系统的分析与综合问题。

根据根轨迹分析系统的方法：

（1）稳定性。当开环增益 K 从零到无穷大变化时，根轨迹不会越过虚轴进入 S 的右半平面，因此这个系统对所有的 K 值都是稳定的。如果根轨迹越过虚轴进入 S 的右半平面，则其交点的 K 值就是临界稳定开环增益。

（2）稳态性能。开环系统在坐标原点有一个极点，因此根轨迹上的 K 值就是静态速度误差系数，如果给定系统的稳态误差要求，则可由根轨迹确定闭环极点容许的范围。

（3）动态性能。当所有闭环极点位于实轴上，系统为过阻尼系统，单位阶跃响应为非周期过程；当闭环两个极点重合，系统为临界阻尼系统，单位阶跃响应仍为非周期过程，但速度更快；当闭环极点为复数极点，系统为欠阻尼系统，单位阶跃响应为阻尼振荡过程，且超调量与 K 成正比。

4．频域分析

频域分析法是应用频率特性研究控制系统的一种典型方法。频率特性是指系统在正弦信号作用下，稳态输出与输入之比对频率的关系特性。从频率响应中可以得出带宽、增益、转折频率、闭环稳定性等系统特征。

频域分析法可直观地表达系统的频率特性，分析方法比较简单，物理概念比较明确，对于诸如防止结构谐振、抑制噪声、改善系统稳定性和暂态性能等问题，都可以从系统的频率特性上明确地看出其物理实质和解决途经。

频率特性函数与传递函数有直接的关系，即

$$G(\mathrm{j}\omega)=\frac{X_{\mathrm{o}}(\mathrm{j}\omega)}{X_{\mathrm{i}}(\mathrm{j}\omega)}=A(\omega)\mathrm{e}^{\mathrm{j}\varphi(\omega)}$$

式中，$A(\omega)=\dfrac{X_{\mathrm{o}}(\omega)}{X_{\mathrm{i}}(\omega)}$ 为幅频特性，$\varphi(\omega)=\varphi_{\mathrm{o}}(\omega)-\varphi_{\mathrm{i}}(\omega)$ 为相频特性。

由于系统的频率特性 $G(\mathrm{j}\omega)$的代数表达式比较复杂，一般采用图形表达方法，常用的图形有波特图、奈奎斯特图和尼科尔斯图。

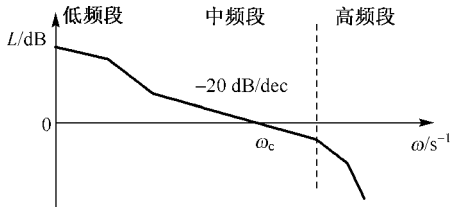


图 7-1-3　典型的控制系统波特图

1）波特图

波特图即对数频率特性图，包括对数幅频特性图和对数相频特性图。横坐标为频率ω，采用对数分度，单位为弧度/秒；纵坐标均匀分度，分别为幅值函数 $20\lg A(\omega)$，dB 表示相角，以度表示。利用对数频率特性图可以展宽频率范围，而且曲线形状简单，容易分析和绘制。

在设计校正装置时，波特图是主要的研究工具，如图 7-1-3 所示，即开环对数频率特性的渐近线。可以确切地提供稳定性和稳定裕度的信息，而且还能大致衡量闭环系统稳态和动态的性能。

在定性地分析闭环系统性能时，通常将波特图分成低、中、高三个频段，频段的分割界限是大致的，从图 7-1-3 中三个频段的特征可以判断系统的性能，这些特征包括以下四个方面：

- 高频段衰减越快，即高频特性负分贝值越低，说明系统抗高频噪声干扰的能力越强。
- 中频段以 20 dB/dec 的斜率穿越零分贝线，而且这一斜率覆盖足够的频带宽度，则系统的稳定性好；
- 低频段的斜率陡、增益高，说明系统的稳态精度高；
- 截止频率（或称剪切频率）越高，则系统的快速性越好；

以上四个方面常常是互相矛盾的。对稳态精度要求很高时，常需要放大系数，但可能使系统不稳定；加上校正装置后，系统稳定但可能牺牲快速性；提高截止频率可以加快系统的响应，但容易引入高频干扰。

设计时往往须在稳、准、快和抗干扰这四个矛盾的方面之间取折中，才能获得比较满意的结果。

波特图中的稳定裕度是衡量最小相位系统稳定程度（即相对稳定性）的重要指标，保留适当的稳定裕度可以防止系统在各元件参数发生变化后导致不稳定，稳定裕度也能间接地反映系统动态过程的平稳性，稳定裕度大意味着振荡弱、超调小。

2）奈奎斯特图

奈奎斯特图是对于一个连续时间的线性非时变系统，将其频率响应的增益及相位以极

坐标的方式绘出，即对于频率特性函数 $G(j\omega)$，给出频率 ω 从负无穷到正无穷的一系列数值，分别求出振幅 $\mathrm{Im}(G(j\omega))$和相位 $\mathrm{Re}(G(j\omega))$。以 $\mathrm{Re}(G(j\omega))$为横坐标，$\mathrm{Im}(G(j\omega))$为纵坐标绘制成极坐标频率特性图。

奈奎斯特图上每一点都是对应一特定频率下的频率响应，可以用来判断一个有反馈的系统是否稳定。

3）尼科尔斯图

尼科尔斯图又称为对数幅相图，是在直角坐标上以频率 ω 为参量表示的对数幅值 $20\log|G(j\omega)|$与相角 $\angle G(j\omega)$的一种关系图。尼科尔斯图很容易根据波特图上的对数幅值特性和相角特性来绘制。尼科尔斯图的优点是能较容易地确定控制系统的相对稳定性。

7.1.7 控制系统的设计

控制系统的设计，就是在系统中引入适当的环节。用以对原有系统的某些性能进行校正，使之达到理想的效果，又称为系统的校正。通常是给系统附加一些具有某种典型环节特性的电网络，运算部件或测量装置等，如无源或有源微积分电路，以及速度、加速度传感器等，称为校正元件或校正装置，靠这些装置的配置来有效地改善整个系统的控制性能。

控制系统的设计过程可以在时域进行，也可以在频域进行。

如果对象模型是以传递函数的形式给出，通常采用经典控制理论中的频率特性法或根轨迹法完成控制器的设计，即在原有系统中引入适当的环节，用以对原有系统的某些性能（如相角裕度、剪切频率、误差系数等）进行校正，改变系统的频率特性形状，使校正后的系统频率特性具有合适的低频、中频和高频特性，以及足够的稳定裕量，从而满足所要求的性能指标。

如果对象模型是在状态空间以状态方程形式描述的，则系统的设计过程是在时域进行的，通常是采用状态反馈和极点配置的方法得到控制策略，其中包括状态观测器的设计以及最优控制系统的设计等。

常用的设计方法有：串联校正、反馈校正、PID 控制器设计、极点配置与状态观测器设计、线性二次型最优控制系统设计等。

1．串联校正

在系统主反馈回路内采用的校正方法，校正装置串联在系统的前向通道中。

控制系统中常用的串联校正装置是带有单零点与单极点的滤波器。若其零点比极点更靠近原点，则称之为超前校正；否则称之为滞后校正。

2．反馈校正

在系统主反馈回路内采用的校正方法，在系统中增加某些局部反馈环节。

3．PID 调节器

在工程实际中，应用最为广泛的调节器控制规律为比例、积分、微分控制，简称 PID 控制。当被控对象的结构和参数不能完全掌握，或得不到精确的数学模型时，控制理论的

其他技术难以采用时，系统控制器的结构和参数必须依靠经验和现场调试来确定，这时应用 PID 控制技术最为方便。PID 控制是最早发展起来的方法之一，由于其结构简单，应用中参数整定方便，因此在工业控制中得到广泛的应用。当今应用的工业控制器中，有半数以上是采用 PID 或变形 PID 控制方案的，其中包括传统的模拟式 PID 控制和近年来微处理器实现的数字 PID 控制器。

4．极点配置与状态观测器设计

经典控制理论通常采用输出反馈，用频率法或根轨迹法设计控制系统，使闭环极点在期望的位置上。现代控制理论则更多地采用状态和极点任意配置的方法进行综合。由于状态反馈可以提供更丰富的状态信息和选择的自由度，因此使系统容易获得更优良的性能。

给定控制系统，通过设计反馈增益 k 使闭环系统具有期望的极点，从而达到适当的阻尼系数和无阻尼自然频率，这就是极点配置问题。但极点配置是基于状态反馈的，即 $\boldsymbol{u} = -\boldsymbol{kx}$，因此状态 $\boldsymbol{x}$ 必须可测，当状态不可测时，则应设计状态观测器。设计的状态观测器也应具有适当的频率特性，因此也可指定其极点的位置，从而使状态观测器的设计转化为极点配置问题。

对系统的状态空间描述，经常由于状态变量选择的非唯一性，得到的状态空间表达式不唯一。在实际应用中，可根据所研究的问题选取相应的状态表达形式。将状态空间表达式化成对角线标准型或约旦标准型，对系统能控性和能观性的分析将十分方便。

5．线性二次型最优控制系统设计

在线性系统最优控制中，有一类最优控制系统，其性能指标泛函是状态变量和控制变量的二次型函数的积分，此类最优控制问题称为线性二次型最优控制问题，简称线性二次型（LQ）。

LQ 问题解出的控制规律是状态变量的线性函数，可以通过状态反馈实现闭环最优控制，因而在工程上得到广泛的应用。

7.2 控制系统的建模

7.2.1 基本数学模型

对于控制系统中的基本数学模型，控制系统工具箱都有相应的函数进行描述，模型之间可以进行转换。

用于控制系统建模及模型转换的函数见表 7-2-1。

表 7-2-1 建模及模型转换函数

函　数	功　能	函　数	功　能
tf	创建或转换为传递函数模型	ss	创建或转换为状态空间模型
zpk	创建或转换为零极点增益模型	c2d	由连续时间模型转换为离散时间模型
c2dm	按照指定方式将连续时间模型转换为离散时间模型	d2cm	按照指定方式将离散时间模型转换为连续时间模型
d2c	由离散时间模型转换为连续时间模型	d2d	离散时间系统重新采样

（续表）

函　数	功　能	函　数	功　能
ss2tf	状态空间模型转换为传递函数模型	ss2zp	状态空间模型转换为零极点增益模型
tf2ss	传递函数模型转换为状态空间模型	tf2zp	传递函数模型转换为零极点增益模型
zp2ss	零极点增益模型转换为状态空间模型	zp2tf	零极点增益模型转换为传递函数模型
residue	传递函数模型与部分分式模型互换	ge	获得 LTI 对象的属性
set	设置和修改 LTI 对象的属性	tfdata	获得变换后的传递函数模型参数
ssdata, dssdata	获得变换后的状态空间模型参数	zpkdata	获得变换后的零极点增益模型参数
class	模型类型的检测	series	系统的串联
parallel	系统的并联	feedback	系统的反馈连接
appen	多个 LTI 系统的组合	—	—

1．传递函数模型

当系统用传递函数表示时，有

$$G(s)=\frac{y(s)}{u(s)}=\frac{b_1s^m+b_2s^{m-1}+\cdots+b_{m+1}}{a_1s^n+a_2s^{n-1}+\cdots+a_{n+1}}$$

```
num = [b1, b2, …, bm+1]
    den = [a1, a2, …, am+1]
sys=tf(num,den)
```

注意：

（1）构成分子、分母的向量按降幂排列的顺序，缺项部分必须用 0 补齐；

（2）如果传递函数的分子、分母均为多项式相乘的形式，不能直接写出，可借助多项式乘法运算函数 conv 来处理，以便获得分子、分母多项式向量；

（3）对离散时间系统的动态模型，用脉冲传递函数描述，其输入方法与此类似。

【例 7-2-1】建立如下传递函数的系统模型。

$$G(s)=\frac{4(s+2)(s^2+6s+6)^2}{s(s+1)^3(s^3+3s^2+2s+5)}$$

程序如下：

```
num=4*conv([1,2],conv([1,6,6],[1,6,6]));
den=conv([1,0],conv([1,1],conv([1,1],conv([1,1],[1,3,2,5]))));
sys1=tf(num,den)
```

对于多输入多输出的 MIMO 系统，则用传递矩阵描述，例如：

$$G(s)=\begin{bmatrix}\dfrac{s+1}{s^2+2s+2}\\[2ex]\dfrac{1}{s+1}\end{bmatrix}$$

可表示为

```
num={[1,1];1}
den={[1 2 2];[1 1]}
sys2=tf(num,den)
```

控制系统常用到并联系统，这时就要对系统函数进行分解，使其表现为一些基本控制单元的和的形式。函数[r,p,k]=residue(b,a)对两个多项式的比进行部分展开，以及把传递函数分解为微分单元的形式。

【例 7-2-2】分解系统为微分单元形式。

$$G(s)=\frac{2s^3+9s+1}{s^3+s^2+4s+4}$$

程序如下：

```
num=[2,0,9,1];
den=[1,1,4,4];
[r,p,k]=residue(num,den)
```

结果为：

```
r = -0.0000 - 0.2500i
    -0.0000 + 0.2500i
    -2.0000
p = -0.0000 + 2.0000i
    -0.0000 - 2.0000i
    -1.0000
k = 2
```

对应的系统表达式为：

$$G(s)=2+\frac{-0.25i}{s-2i}+\frac{0.25i}{s+2i}+\frac{-2}{s+1}$$

2．零极点模型

对于零极点描述的 SISO 系统：

$$G(s)=\frac{k(s+z_1)(s+z_2)(s+z_3)\cdots(s+z_m)}{(s+p_1)(s+p_2)(s+p_3)\cdots(s+p_n)}$$

先用向量的形式输入系统的零点和极点增益，然后调用 zpk 函数。

```
z=[z1 z2…zm-1 zm];
p=[p1 p2…pn-1 pn];
k=k;
sys=zpk(z,p,k)
```

如果已知传递函数形式，分别对分子和分母做因式分解，则可以得出系统的零极点模型。这可以通过求出分子和分母多项式的根 roots 函数来实现。

对于多输入多输出系统，应分别对每个输入求出系统的零极点模型，最后才可以获得整个系统的零极点模型（为矩阵形式）。

【例 7-2-3】已知系统传递函数，求零极点模型。

$$G(s)=\frac{2s^3+9s+1}{s^3+s^2+4s+4}$$

程序如下：

```
num=[2,0,9,1];
den=[1,1,4,4];
[z,p,k]=tf2zp(num,den);
sys2=zpk(z,p,k)
```

3．状态方程模型

LTI 系统的状态方程：

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}$$

只要将 $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}$ 几个矩阵输入进去即可，对于离散系统，也与上面类似。

【例 7-2-4】用 ss 模型描述两输入两输出系统：

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 1 & 6 & 9 & 10 \\ 3 & 12 & 6 & 8 \\ 4 & 7 & 9 & 11 \\ 5 & 12 & 13 & 14 \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 4 & 6 \\ 2 & 4 \\ 2 & 2 \\ 1 & 0 \end{bmatrix}\boldsymbol{u}$$

$$\boldsymbol{y} = \begin{bmatrix} 0 & 0 & 2 & 1 \\ 8 & 0 & 2 & 2 \end{bmatrix}\boldsymbol{x}$$

程序如下：

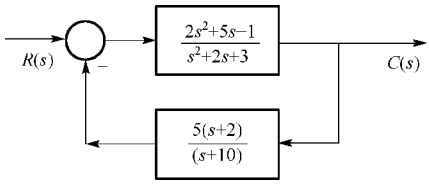
```
A=[1 6 9 10; 3 12 6 8; 4 7 9 11; 5 12 13 14];
B=[4 6; 2 4; 2 2; 1 0];
C=[0 0 2 1; 8 0 2 2];
D=zeros(2,2);
sys3=ss(A,B,C,D)
```

4．结构图模型

采用结构图方式描述的系统，已知各模块的数学函数，需对系统进行连接组合。连接组合方式主要有串联、并联、反馈等形式。主要的连接函数有

- series：系统串联实现；
- parallel：系统并联实现；
- feedback：系统反馈连接；
- append：多个 LTI 系统的组合。

【例 7-2-5】计算如下系统的传递函数。



程序如下：

```
s1=tf([2,5,1],[1,2,3])      %系统 s1 的传递函数模型
s2=zpk(-2,-10,5)            %系统 s2 的零极点增益模型
sys=feedback(s1,s2)         %s1 环节前向，s2 环节反馈
```

5．LTI 对象

为了避免对一个系统采用多个分离变量进行描述，采用 tf、zpk、ss 函数的返回值均为一个 LTI 对象。在控制系统工具箱中，有三种对象，即 ss 对象、tf 对象和 zpk 对象。MATLAB 中很多分析函数中系统的描述既可用独立参数方式也可用 LTI 表示。例如，时域分析中阶跃响应函数 step 可由多种形式描述。

- step（num,den）：num，den 分别为系统传递函数的分子、分母多项式；
- step（A,B,C,D）：A，B，C，D 为状态空间系数矩阵；
- step（z,p,k）：z，p，k 分别为零点、极点及增益；
- step（sys）：其中 sys 可以是用 tf、ss 或 zpk 三个函数中的某一个所建立的 LTI 对象。

为简化说明，在本书后面各章节函数格式的描述均采用 LTI 对象 sys 描述方式。

函数 tf、zpk 和 ss 将模型参数与采样周期封装到一个单独的 LTI 对象中以简化其引用，反过来控制系统工具箱也提供了一些函数来得到这些参数。

例如：

```
[num,den,Ts]=tfdata(sys )          %Ts 为采样时间
[z,p,k,Ts]=zpkdata(sys)
[a,b,c,d,Ts]=ssdata(sys)
```

此外，还可用函数 get(sys)、set(sys)获取和设置 sys 的属性。例如：

```
get(sys)
num: {[0 1]}
den: {[1 2]}
Variable: 's'
Ts: 0
ioDelay: 0
InputDelay: 0
OutputDelay: 0
InputName: {''}
OutputName: {''}
InputGroup: {0x2 cell}
OutputGroup: {0x2 cell}
Notes: {}
```

【例 7-2-6】系统的传递函数为

$$H(s)=\frac{2s^2+5s+1}{s^2+2s+3}$$

输入延时 $T_d=0.35$ s，试用一阶保持法对连续系统进行离散，采样周期 $T_s=0.1$ s。

程序如下：

```
sys=tf([2,5,1],[1,2,3],'td',0.5);      %生成连续系统的传递函数模型
sysd=c2d(sys,0.1,'foh')                %形成采样系统
```

7.2.2 模型的转换

在进行系统分析时，往往根据不同的要求选择不同形式的数学模型，因此经常要在不同形式的数学模型之间相互转换。

1．连续系统模型间的转换

对于连续系统，仿真中通常将微分方程作为描述动态性能的基本形式，作为共性的内容进行分析时，又常常将其转换为传递函数形式，在计算机中，利用系统的状态空间描述最方便。

【例 7-2-7】已知系统传递函数如下所示，将模型转换为状态方程模型和零极点增益模型。

$$G(s)=\frac{6s^2+42s+72}{s^3+6s^2++11s+6}$$

程序如下：

```
num=[0 6 42 72];
den=[1 6 11 6];
[A,B,C,D]=tf2ss(num,den)
[z,p,k]=tf2zp(num,den)
```

程序执行后得到的状态方程模型为

$$\begin{cases}\dot{\boldsymbol{x}}(\boldsymbol{t})=\begin{bmatrix}-6 & -11 & -6\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\boldsymbol{x}(\boldsymbol{t})+\begin{bmatrix}1\\0\\0\end{bmatrix}\boldsymbol{u}(\boldsymbol{t})\\ \boldsymbol{y}(\boldsymbol{t})=[6\ \ 42\ \ 72]\boldsymbol{x}(\boldsymbol{t})\end{cases}$$

零极点增益模型 $G(s)$为

$$\frac{6(s+3)(s+4)}{(s+1)(s+2)(s+3)}$$

【例 7-2-8】已知系统的部分分式如下所示，求其传递函数模型。

$$G(s)=2+\frac{-0.25i}{s-2i}+\frac{0.25i}{s+2i}+\frac{-2}{s+1}$$

程序如下：

```
r=[-0.25i,0.25i,-2];
p=[2i,-2i,-1];k=2;
[num,den]=residue(r,p,k)
```

注意：余式一定要与极点相对应。

2．连续域与离散域的转换

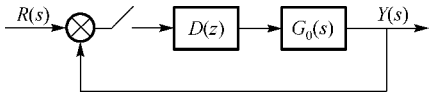
控制系统工具箱中提供了连续域与离散域的相互转化的函数调用。如函数 c2d 可将连续系统离散化，相反 d2c 将离散系统向连续域转化。工具箱支持常用的几种转化方法，包括带零阶保持器的离散化方法、带一阶保持器的离散化方法、Tustin 变换和带预修正的 Tustin 变换以及零极点匹配法。

例如，连续系统的离散化函数 sysd=c2d(sysc,'fp')。

sysc 为连续系统的数学模型，sysd 为离散系统的数学模型，选项 fp 的具体内容为

- zoh：零阶保持器法；
- foh：一阶保持器法；
- imp：脉冲响应不变法；
- tustin：双线性变换法；
- prewarp：改进的双线性变化法；
- matched：零极点匹配法。

【例 7-2-9】控制系统结构如下所示，已知各环节传递函数，试绘制系统的闭环阶跃响应图。



$$G_0(s)=\frac{0.1}{s(s+0.1)},\qquad D(z)=\frac{9.1544(z-0.98)}{z-0.8187},\qquad T=0.2$$

程序如下：

```
clear;
num=0.1;
den=[1 0.1 0];
G0s=tf(num,den)                    %G0(s)的传递函数
Z=[0.98];
P=[0.8187];
K=9.1544;
Dz=zpk(Z,P,K,'Ts',0.2)             %控制器 D(z)
G0z=c2d(G0s,0.2,'zoh')             %G0z=Z(Gh(s)*G0(s))
Gz=Dz*G0z                          %开环传递函数
faiz=feedback(Gz,1);               %闭环传递函数
step(faiz)                         %t 表示仿真时间
```

闭环阶跃响应曲线如图 7-2-1 所示。

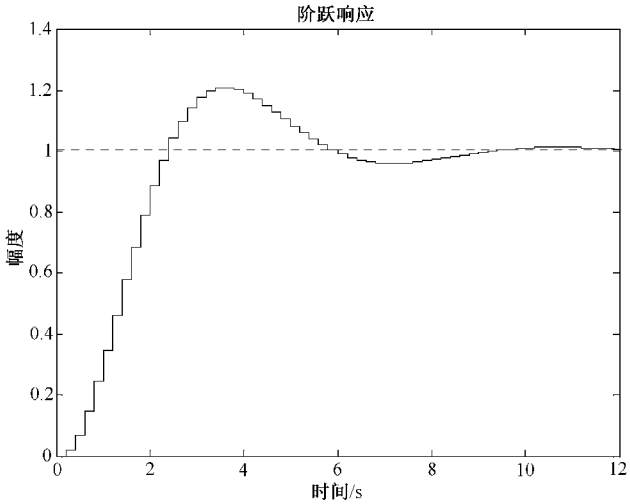


图 7-2-1　闭环阶跃响应曲线

7.2.3 模型的属性描述、降阶与标准化

MATLAB 提供了许多用来分析控制系统模型属性的函数，如常用的可控性、可观性、阻尼系数、自然频率等。

在系统的研究中，模型降阶技术有很重要的作用，模型降阶技术的基本思想是使原始系统的系数矩阵的阶次降低，并保留原系统的主导特征值和一些重要的状态。为分析设计高阶系统带来方便。

在 MATLAB 控制系统工具箱中提供了基于平衡实现降阶函数 balreal 和 modred。函数 balreal 计算可控及可观测的 Gram 矩阵，并对原系统进行等价变换，将原系统分成两部分，其中一部分包含原系统矩阵较大奇异值，而另一部分包含较小奇异值，平衡实现系统完全与原系统等价。

【例 7-2-10】已知系统的状态空间表达式为

$$\dot{\boldsymbol{x}}=\begin{pmatrix}0 & 6 & -5\\ 1 & 0 & 2\\ 3 & 2 & 4\end{pmatrix}\boldsymbol{x}+\begin{pmatrix}5\\ 1\\ 5\end{pmatrix}\boldsymbol{u}$$

$$\boldsymbol{y}=(1\quad 1\quad 2)\boldsymbol{x}$$

试判定系统的能控性和能观性。

程序如下：

```
A=[0 6 -5;1 0 2;3 2 4]
B=[5;1;5]
C=[1 1 2]
Qc=ctrb(A,B)
n=rank(Qc)
if(n==3)
    disp('system is controllable')
else
    disp('system is uncontrollable')
end
Qo=obsv(A,C)
m=rank(Qo)
if(m==3)
    disp('system is observable')
else
    disp('system is unobservable')
end
```

【例 7-2-11】连续系统四阶模型为

$$h(s)=\frac{s^3+11s^2+36s+26}{s^4+14.6s^3+74.96s^2+153.7s+99.65}$$

试对该系统进行降阶处理。

程序如下：

```
h=tf([1 11 36 26],[1 14.6 74.96 153.7 99.65])
[hb,g]=balreal(h)
g'
hmdc=modred(hb,2:4,'MatchDC')
hdel=modred(hb,2:4,'Truncate')
```

7.2.4 延迟系统模型

针对带有时间延迟的控制系统传函模型，其中 T 为延迟时间常数。其数学描述为

$$G(s)=\frac{b_1s^m+b_2s^{m-1}+\cdots+b_ms+b_{m+1}}{a_1s^n+a_2s^{n-1}+\cdots+a_1s+a_{n+1}}\mathrm{e}^{-Ts}=\hat{G}(s)\mathrm{e}^{-Ts}$$

纯时间滞后环节可以由有理函数来近似，1892 年法国数字家 Pade 曾提出了一种著名的有理近似方法，后人将它命名为 Pade 近似法。其表达式为

$$\mathrm{e}^{-Ts}=\frac{1-Ts/2+P_1(Ts)^2-P_2(Ts)^3+P_3(Ts)^4-P_4(Ts)^5+\cdots}{1+Ts/2+P_1(Ts)^2+P_2(Ts)^3+P_3(Ts)^4+P_4(Ts)^5+\cdots}$$

控制系统工具箱提供了函数 pade 可以求取 Pade6 近似的有理传递函数模型，函数的调用格式为

```
[np,dp]=pade (Tau,n)
```

其中，Tau 为延迟常数，n 为 Pade 阶次的阶次。Pade 有理近似模型的分子和分母在（np,dp）变量中返回。

建立延迟系统的数学模型还有以下方法：

（1）用函数 tf、zpk 或 ss 建模并同时给模型时间延迟属性“IoDelayMatrix、InputDelay、OutputDelay”赋值。

（2）先用函数 tf、zp 或 ss 建模，再用 set 命令给模型时间延迟属性赋值。

（3）用 Simulink 的 Transport delay 模块描述延迟。

【例 7-2-12】已知系统传递函数如下，用 MATLAB 生成系统模型程序。

$$H(s)=\frac{2}{s+1}\mathrm{e}^{-0.1s}$$

程序如下：

```
num=2;
den=[1 1];
sys=tf(num,den,'InputDelay',0.1)
```

或

```
num=2;
```

```
den=[1 1];
sys=tf(num,den)
set(sys,'InputDelay',0.1)
```

7.2.5 非线性系统的模型

在工程实际中，由于组成控制系统各元件的动态和静态特性都存在着不同程度的非线性，因此理想的线性系统并不存在。这种系统一般很难列出整个系统的微分方程组，只能按照系统中各个环节的联结情况安排出顺序，对每个环节按次序分别进行计算，因此有一个计算次序的排列问题。非线性系统的建模可以通过以下方法实现。

（1）简单的典型非线性环节可以用 Simulink 现有模块搭建。

（2）复杂的非线性环节无法等效变换，可以考虑用查表模块实现，找出转折点坐标，写入查表模块。

（3）将非线性系统线性化处理。非线性系统的线性化，通常是采取把非线性系统在某一平衡点附近线性化的方法，利用线性化的模型进行分析。Simulink 提供了函数 linmod、dlinmod 对连续和离散系统进行模型的线性化。它们的用法相似，linmode 的调用格式如下：

```
[A,B,C,D]=linmod ('sys',x,u,pert,xpert,upert)
```

函数说明：

（1）函数的 sys 参数必不可少，它是 Simulink 模型的名字。

（2）可使用 x,u 来设定系统的状态和输入工作点，默认值全为零。

（3）当模型是非线性时，就应选择一个运算点来提取线性模型。非线性模型对发生在提取模型点的扰动大小也是敏感的。必须在各种舍入与截位误差之间选择一种折中方案。linmod 命令中的额外参数用于指定运算点和扰动点。

linmod2 函数要比 linmod 精确，但运算时间要长。

模型线性化处理，必须要取得一个平衡工作点，然后在平衡工作点附近进行线性化。平衡点，又称为 trim 点，是处于稳定状态的动态系统的参数空间的点。用数学方法描述 trim 点是系统状态导数为零的点。trim 函数从一起始点开始搜索，使用一个顺序二次程序设计算法，直到找到最近的 trim 点为止。用户必须隐含或明示地提供起始点，如果 trim 函数不能找到 trim 点，则返回搜索时遇见的状态微分在最小或最大意义上的最接近零的点，也就是返回使微分对零的最大偏移为最小值时的点。对于非线性系统，在不同的平衡点附近数学模型有较大的不同。trim 函数的调用格式为

```
[x,u,y,dx]=trim ('sfun',x0,u0,y0)
```

其中，sfun 为模型文件名，x0 为初始假设，u0、y0 分别为输入和输出值。

MATLAB 通过寻优的方式寻找最佳平衡点。多次使用不同的初值寻找平衡点是有必要的。一旦得到了线性化的系统状态空间形式描述的模型，就可以进行系统分析，如画 Bode 图，做阶跃响应，判断稳定性等，还可以进行控制系统的各种设计。

7.3 控制系统的分析

7.3.1 稳定性分析

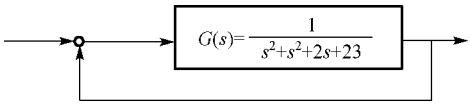
从前面的章节可知，只要知道了系统模型，不论哪种形式的数学模型，都可以很方便地由 MATLAB 求出系统的零极点，从而判断系统的稳定性，这一点对于工程实际而言是十分重要的。MATLAB 提供了多种稳定性判别的方法。

（1）利用 pzmap 函数绘制连续的零、极点图。

（2）利用 tf2zp 函数求出系统的零、极点，从而判断系统的稳定性。

（3）对于状态方程模型，由 eig(***A***)可以求出 ***A*** 阵的特征值；可用多项式求根函数 roots 直接求出特征方程的根。

【例 7-3-1】采用多种方法判断如下系统的稳定性：



程序如下：

```
%首先给出闭环系统的传递函数;
num=1;den=[1 1 2 23];
sys=tf(num,den)
sys1=feedback(sys,1);
%以下可用多种方法判稳;
%方法 1
roots(sys1.den{1})
%方法 2
sys2=zpk(sys1);
sys2.P{1}
%方法 3
sys2=zpk(sys1);
pzmap(sys2)
%方法 4
sys3=ss(sys1);
eig(sys3.a)
%方法 5
[z,p,k]=tf2zp(sys1.num{1},sys1.den{1})
i=find(real(p)>0);
j=length(i);
if j>0
   disp('system is Unstable')
else
   disp('system is Stable');
end
```

以上 5 种方法得出共同结论：系统的特征根的实部有正值，系统不稳定。

【例 7-3-2】已知单位负反馈系统的开环传递函数 $G(s)$，分别计算在典型输入信号 $r(t)=1(t)$、t、$\frac{1}{2}t^2$ 下的稳态误差终值。

$$G(s)=\frac{7(s+1)}{s(s+3)(s^2+4s+5)}$$

系统稳态误差终值为

$$e_{sr}=\lim_{t\to\infty}e_r(t)=\lim_{s\to0}sE_r(s)=\lim_{s\to0}s[R(s)-C(s)]=\lim_{s\to0}sR(s)[1-\phi(s)]$$

在 MATLAB 中，函数 dcgain 可求取系统给定误差的终值。

程序如下：

```
num1=[7 7];     %计算 1-φ(s)的值
den1=[conv(conv([1 0],[1 3]),[1 4 5])];
G=tf(num1,den1);
GG=feedback(G,1,-1);
GGG=tf(GG.den{1}-GG.num{1},GG.den{1});
num2=[1,0];
den2=1;
G1=tf(num2,den2);
GGGG=GGG*G1;
num3=1;             %计算 r(t)=1(t) 时的稳态误差终值 e1
den3=[1 0];
R1=tf(num3,den3);
e1=dcgain(GGGG*R1)
num4=1;             %计算 r(t)=t 时的稳态误差终值 e2
den4=[1 0 0];
R2=tf(num4,den4);
e2=dcgain(GGGG*R2)
num5=1;             %计算 r(t)=1/2 t2 时的稳态误差终值 e3
den5=[1 0 0 0];
R3=tf(num5,den5);
e3=dcgain(GGGG*R3)
```

运行结果为

```
e1=
0
e2=
2.  1429
e3=
Inf
```

7.3.2 时域分析

1．时域响应曲线绘制

MATLAB 提供了若干典型输入响应的函数，时域特性曲线的绘制和性能指标的求取通过简单的函数调用即可实现。

工具箱中的常用时域分析函数见表 7-3-1。

表 7-3-1　时域分析函数

函　数　名	功　　能
impulse(sys)	绘制系统 sys 的单位冲激响应
initial(sys,x0)	绘制状态空间模型 sys 在初始条件 x0 下的零输入响应
lsim(sys,u,T)	绘制 LTI 模型 sys 在任意输入 u、持续时间 T 的作用下的输出
gensig(type,tau)	生成特定的激励信号
step(sys)	绘制系统 sys 的阶跃响应

函数说明：

（1）MATLAB 中控制系统绘图函数均有直接绘图或不绘图返回响应结果参数两种格式。

（2）函数中可输入多组参数，同时仿真多个系统。

【例 7-3-3】已知典型二阶系统的传递函数为

$$G(s)=\frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}$$

其中，$\omega_n=6$，绘制系统在 ζ=0.1, 0.2,…, 1.0, 2.0 时的单位阶跃响应。

程序如下：

```
wn=6;
kosi= [0.1,0.2,1.0,2.0] ;
figure(1)
hold on
for kos=kosi
    num=wn.^2;
    den=[1,2*kos*wn,wn.^2] ;
    step(num,den);
end;
title('Step Response');
hold off
```

二阶系统的阶跃响应曲线如图 7-3-1 所示。

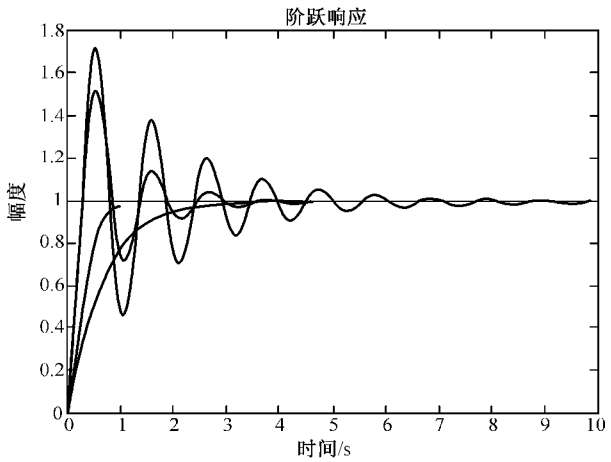


图 7-3-1　二阶系统的阶跃响应

从图 7-3-1 中可以看出，临界阻尼响应具有最短的上升时间，响应速度最快；在欠阻尼的响应曲线中阻尼系数越小，超调量越大，上升时间越短，通常取 $\zeta = 0.4 \sim 0.8$。

【例 7-3-4】已知系统传递函数为

$$G(s) = \frac{4}{s^2 + 2s + 4}$$

求脉冲响应并计算误差。

程序如下：

```
num=[4];den=[1 1 4];
impulse(num,den);
[y,x,t]=impulse(num,den);
trapz(t',y)     %计算误差面积
ans =
0.9983
```

脉冲响应曲线如图 7-3-2 所示。

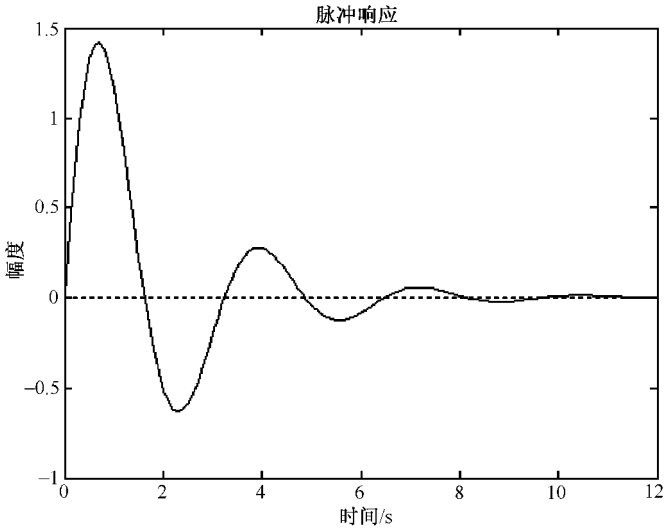


图 7-3-2　脉冲响应曲线

【例 7-3-5】求系统的方波响应，其中方波周期为 6 s，持续时间为 12 s，采样周期为 0.1 s。

$$G(s) = \frac{s+1}{s^2 + 2s + 5}$$

程序如下：

```
[u,t]=gensig('square',6,12,0.1);   %生成方波信号
plot(t,u,'--');hold on;            %绘制激励信号
sys=tf([1,1],[1,2,5]);             %生成传递函数模型
lsim(sys,u,t,'k');                 %系统对方波激励信号的响应
```

系统的方波响应曲线如图 7-3-3 所示。

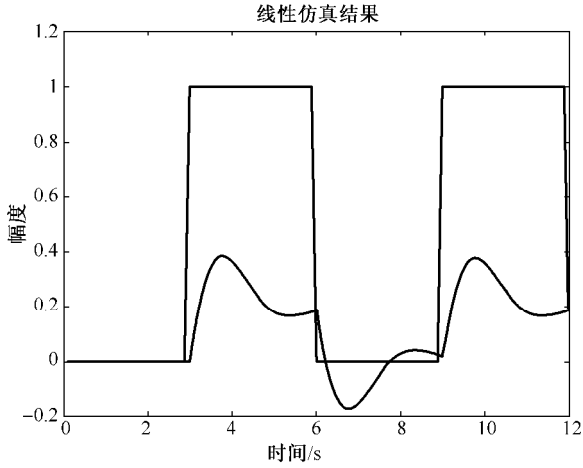


图 7-3-3　系统的方波响应

2．动态性能指标分析

在控制理论中介绍典型线性系统的阶跃响应分析时通常用一些指标来描述，如系统的超调量、上升时间、调节时间等，在 MATLAB 自动绘制的系统阶跃响应曲线上，单击曲线上的某点，可以显示该点对应的时间信息和响应的幅值信息，或者单击鼠标右键，在菜单中，选择其中的 Characteristics 子菜单，从中选择合适的分析内容，可得出系统的阶跃响应指标。

如需要得到更多指标参数，可通过编程实现，把常用的分析程序通过自编函数来实现。

【例 7-3-6】已知系统传递函数为

$$G(s)=\frac{4}{s^2+2s+4}$$

求阶跃响应，并作系统性能分析。

程序如下：

```
num=[4];den=[1 1 4];
[y,x,t]=step(num,den);
tp=spline(y,t,max(y))    %spline 为三次样条插值函数
tp =
1.6062
```

系统的阶跃响应及峰值时间如图 7-3-4 所示。

【例 7-3-7】自定义函数 steppa，函数格式为

```
[mp,tp,sigma,yss]=steppa (y,t)
```

其中，输入参数为 y，t。t 为时间矢量，y 为线性系统单位阶跃响应的输出矢量，由[y,t]=step（sys,t）获得。计算的各性能参数为

- mp：阶跃信号的最大值；
- tp：阶跃信号最大值的对应时间；
- sigma：阶跃响应的超调量；

- yss：阶跃响应的稳态值。

程序如下：

```
function [mp,tp,b1,sigma]=steppa(y,t)
[mp,tf]=max(y);
tp=t(tf);
ct=length(t);
tm=max(t);
m=1
yss=y(ct);
sigma=100*mp/yss
```

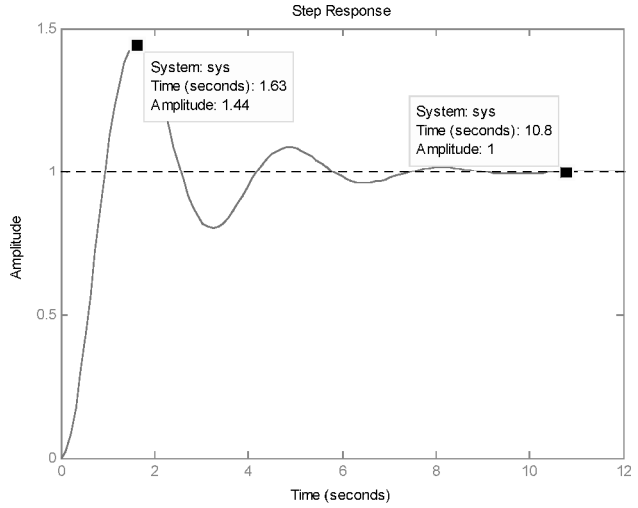


图 7-3-4　系统的阶跃响应及峰值时间

7.3.3　根轨迹分析

根轨迹是指当系统开环的某一个（或几个）参数（一般取系统的开环放大倍数 K）从零到正无穷时，闭环特征方程的根在复平面上的轨迹。由于控制系统的动态特性是由系统闭环零极点共同决定的，而控制系统的稳定性又是由闭环系统极点的位置唯一决定的，因此，在分析控制系统动态性能时，确定闭环系统的零极点在 s 平面上的位置就可以了解系统的基本特性。

有关根轨迹绘制及零极点分析函数见表 7-3-2。

表 7-3-2　根轨迹绘制及零极点分析函数

函 数 格 式	功　　能
pzmap(sys)	绘制系统的零极点图
z = tzero(sys)	求系统的传输零点
[K,poles] = rlocfind(sys)	计算给定根轨迹的增益和极点
rlocus(sys)	绘制系统的根轨迹
[Wn,Z] = damp(sys)	求系统极点的固有频率和阻尼系统

（续表）

函 数 格 式	功　　能
p = pole(sys)	求系统的极点
k = dcgain(sys)	求系统的直流（稳态）增益
s = dsort(p)	离散系统极点按幅值降序排列
s = esort(p)	连续系统极点按实部降序排列
sgrid	在连续系统根轨迹图和零极点图中绘制出阻尼系数和自然频率栅格

函数说明：

（1）pzmap(sys)直接在 s 复平面上绘制出系统对应的零极点位置，极点用×表示，零点用 o 表示。

（2）rlocus(sys)根据 SISO 开环系统 sys 的模型，直接在屏幕上绘制出系统的根轨迹图。开环增益的值从零到无穷大变化。

R=rlocus(sys,k)，[r,k]=rlocus(sys)则根据开环增益变化矢量 k，返回闭环系统特征方程 1+k×num(s)/ den(s)=0 的根 r，它有 length(k)行，length(den)–1 列，每行对应某个 k 值时的所有闭环极点。或者同时返回 k 与 r。若给出传递函数描述系统的分子项 num 为负，利用 rlocus 函数绘制的是系统的零度根轨迹（正反馈系统或非最小相位系统）。

（3）rlocfind 找出给定的一组根（闭环极点）对应的根轨迹增益。其用法如下：

```
[k,poles]=rlocfind(sys)
```

在 LTI 对象的根轨迹图中显示出十字光标，当用户选择其中一点时，其相应的增益由 k 记录，与增益相关的所有极点记录 poles 中。若要使用该函数，必须首先在当前窗口上绘制系统的根轨迹。

```
[k,poles]=rlocfind(sys,p)
```

定义要得到增益的根矢量 P，即事先给出极点。除了显示出该根对应的增益以外，还显示出该增益对应的其他根。

（4）栅格线绘制函数 sgrid 和 zgrid。sgrid 在连续系统根轨迹或零极点图中绘制栅格线。栅格线由等阻尼系数和等自然频率线构成，阻尼系数 ξ 步长为 0.1，范围从 0 到 1；自然频率ω_n步长为 1 弧度/秒，范围从 0 到 10。在绘制前当前窗口必须包含根轨迹图。

Sgrid(z,ω_n)：自己定义想要绘制的阻尼系数向量 z 和自然频率向量ω_n的范围。在画出的根轨迹图上面单击，可以给出对应于所单击点的极点值、对应增益、阻尼比和超调量等。

zgrid 函数使用方法与 sgrid 函数的类似。

【例 7-3-8】已知系统的开环传递函数为

$$G(s)H(s)=\frac{K(s+2)(s+3)}{s(s+1)}$$

试绘制出闭环系统的根轨迹，并求出当系统闭环极点 p = [–0.707，–2.6]时，系统所对应的增益 K 值。

程序如下：

```
z=[-2 -3];
p=[0 -1];
k=1;
g=zpk(z,p,k);
rlocus(g);
[k,ploes]=rlocfind(g,[-0.707,-2.6])
```

程序执行结果为

```
k=
0.0699   17.3333
ploes=
-0.7070 -2.6000
-0.5542 -2.1818
```

根轨迹如图 7-3-5 所示。

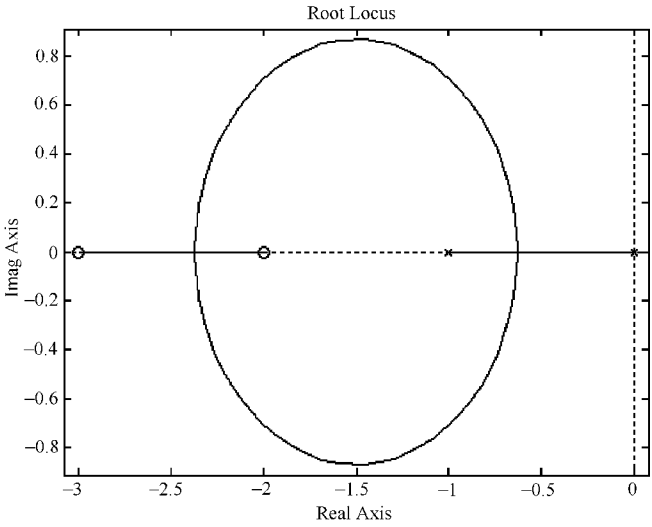


图 7-3-5　根轨迹图

7.3.4　频域分析

稳定的线性定常系统，在正弦输入信号作用下，其输出的稳态分量是与输入同频率的正弦函数。进入稳态以后，输出信号的振幅和输入信号振幅之比称为幅频特性。而输出信号的相位和输入信号的相位之差称为相频特性。MATLAB 提供了 3 种频率响应曲线的绘制函数：Bode 图、Nyquist 曲线及 Nichols 曲线，它们可用于 SISO 或 MIMO 的连续时间和离散时间系统。

在频率分析法中，判别闭环系统稳定性的最基本定理是 Nyquist 判据。对于开环稳定的系统来说，开环传递函数的极点全部位于左半复平面以内，闭环系统为稳定的充分必要条件为：开环频率特性的奈氏曲线不包围（–1，j0）点。在半对数坐标上，分别绘制对数幅频特性和相频特性，称为波特图。

频域分析函数见表 7-3-3。

表 7-3-3　频域分析函数

函 数 格 式	功　　能
bode(sys)	绘制 Bode 图
nichols(sys)	绘制 Nichols 图
Nyquist(sys)	绘制 Nyquist 图
Sigma(sys)	绘制系统奇异值 Bode 图
fresp = evalfr(sys,x)	计算系统单个复频率点的频率响应
dbode(a,b,c,d,Ts,iu)	绘制离散系统的 Bode 图
dnichols(num,den,ts)	绘制离散系统的 Nichols 图
dnyquist(num,den,ts)	绘制离散系统的 Nyquist 图
ngrid	绘制 Nichols 网格图
[gm,pm,wcg,wcp]=margin(sys)	计算增益和相位裕度以及对应的频率
h = freqresp(sys,w)	计算系统在给定实频率区间的频率响应

函数说明：

（1）[mag,pha,w]=bode（sys,iu,w）

sys 为用 tf,ss,zpk 描述的 LTI 模型或各模型的系数描述。iu,w 可选。iu 用来在多输入输出时指明输入变量的序号。w 为给出的频率范围，一般由 w=logspace（a,b,n）给出。缺省时可由系统自动地选择一个合适的频率范围。可得到系统波特图相应的幅值 mag、相角 pha 及角频率点 w 矢量。

（2）[re,im,w]=nyquist（sys）

绘制出系统的一组 Nyquist 曲线，每条曲线相应于连续状态空间系统[A,B,C,D]的输入/输出组合对。返回系统频率特性函数的实部 re 和虚部 im 及角频率点 w 矢量（为正的部分），可以用 plot（re,im）绘制出对应 w 从负无穷到零变化的部分。

（3）[mag,phase,w]=nichols（sys,w）

计算系统的尼科尔斯频率响应曲线，可用于分析开环和闭环系统的特性，其中 mag 为幅值，phase 为相位，w 为角频率点。

（4）[gm,pm,wcg,wcp]=margin（mag,phase,w）

margin 函数可以从频率响应数据中计算出幅值裕度、相角裕度以及对应的频率。幅值裕度和相角裕度是针对开环 SISO 系统而言，它指示出系统闭环时的相对稳定性。当不带输出变量引用时，margin 可在当前图形窗口中绘制出带有裕量及相应频率显示的 Bode 图，其中幅值裕度以分贝为单位。有返回值时，由幅值 mag、相角 phase 及角频率 w 矢量计算出系统幅值裕度 gm、相角裕度 pm 及相应的相角交界频率 wcg、截止频率 wcp。

（5）fh=freqs（num,den,w）

用于计算模拟滤波器的幅频响应，其中实矢量 w 用于指定频率值，返回值 h 为一个复数行向量，要得到幅值必须对它取绝对值，即求模。

【例 7-3-9】已知系统的开环传递函数为

$$G(s)=\frac{10}{s(5s+1)(10s+1)}$$

试绘制其 Bode 图、奈氏曲线，并得出系统频率响应的性能指标。

程序如下：

```
num=10;
den=conv([1 0],conv([5,1],[10,1])) ;
G=tf(num,den);
bode(G,{0.001,100})
margin(G)
[Gm,Pm,Wcg,Wcp]=margin(G)
title('BodeDiagramofG(s)=10/[s(5s+1)(10s+1)]')
figure(2)
nyquist(G,{0.1,100})
title('NyquistDiagramofG(s)=10/[s(5s+1)(10s+1)]')
```

程序执行结果如图 7-3-6 和图 7-3-7 所示。

```
Gm=          %幅值裕度
0.0300
Pm=          %相角裕度
-60.7504
Wcg=
0.1414
Wcp=         %截止频率
0.5707
```

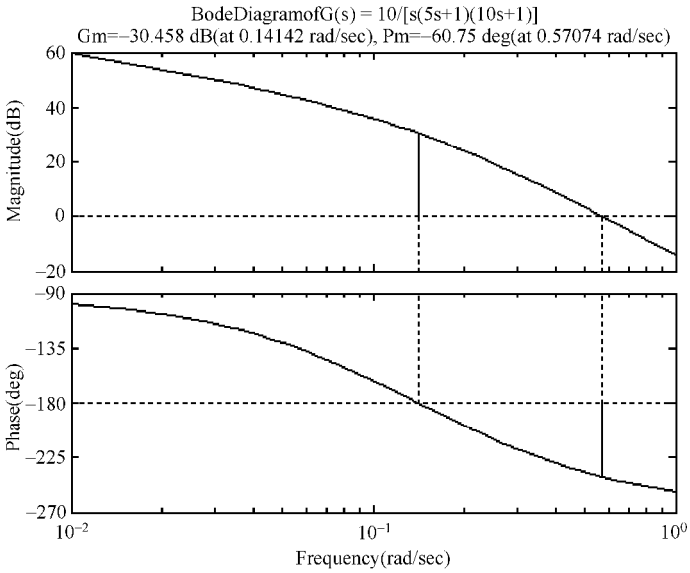


图 7-3-6　开环系统 Bode 图及频率响应参数

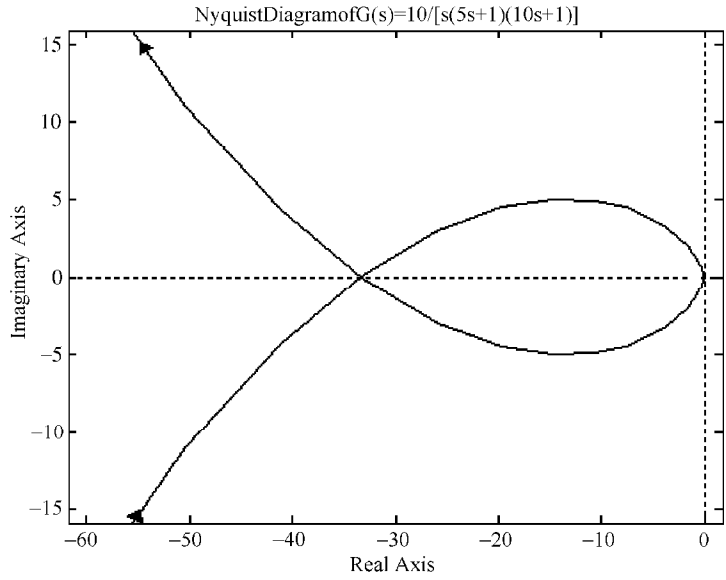


图 7-3-7　开环系统奈氏曲线图

【例 7-3-10】试绘制开环系统 $H(s)$的 Nyquist 曲线，判断闭环系统的稳定性，并求出闭环系统的单位冲激响应。

$$H(s)=\frac{50}{(s+5)(s-2)}$$

程序如下：

```
k=50;z=[];p=[-5,2];
sys=zpk(z,p,k);
figure(1);nyquist(sys);title('Nyquist 曲线图');
figure(2);sb=feedback(sys,1);
impulse(sb);title('单位冲激响应');
```

开环系统的 Nyquist 曲线及冲激响应如图 7-3-8 所示。

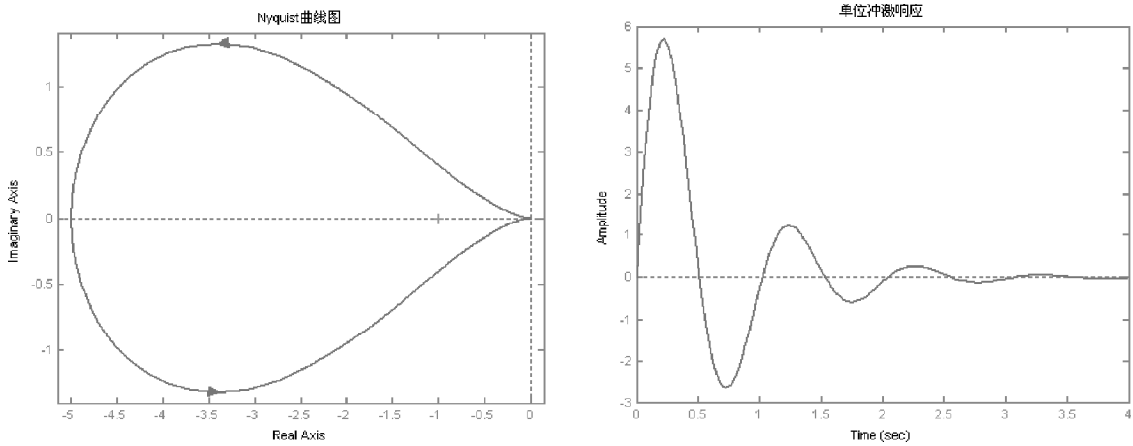


图 7-3-8　开环系统的 Nyquist 曲线图及冲激响应

【例 7-3-11】已知系统的开环模型为

$$G(s)=\frac{k}{s(s+1)(s+2)}$$

当 $k=2$，$k=10$ 时，分别绘制尼科尔斯图线，并作尼科尔斯图线分析。

程序如下：

```
n=[2];
d=[1 3 2 0];
ngrid('new')
nichols(n,d)
hold on
n=[10];
nichols(n,d)
```

由图 7-3-9 可知，$k=2$ 时，闭环系统约有 6 dB 的闭环谐振峰值；$k=10$ 时，曲线已经切过无穷大点，因此系统是不稳定的。

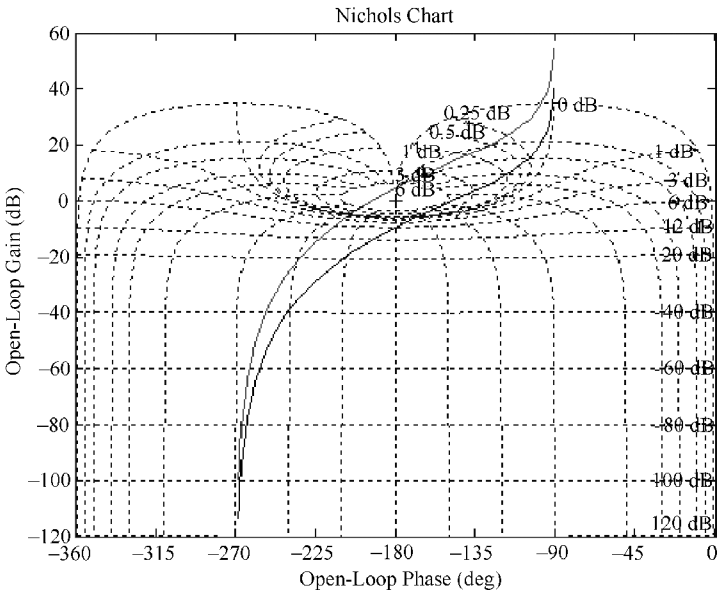


图 7-3-9　尼科尔斯图线

7.3.5　系统分析工具 ltiview

控制系统工具箱提供了几个 GUI 工具供用户使用，主要有 ltiview、sisotool、rltool 等，其中 ltiview 用于分析 LTI 系统，并绘制各种响应图。

输入 ltiview 命令打开一个系统分析窗口界面，用于 LTI 系统的时域分析或者频域分析，它将用户经常进行的一些系统分析功能集中到了一个界面友好、使用方便的对话式窗口中，这样用户就不必一条一条地在命令行下输入相关的函数。

在菜单中选择需分析的系统导入，即可分析此系统。

在图形界面上通过菜单项的选取可选择分析曲线类型、线型等，内容很丰富，如图 7-3-10 所示。

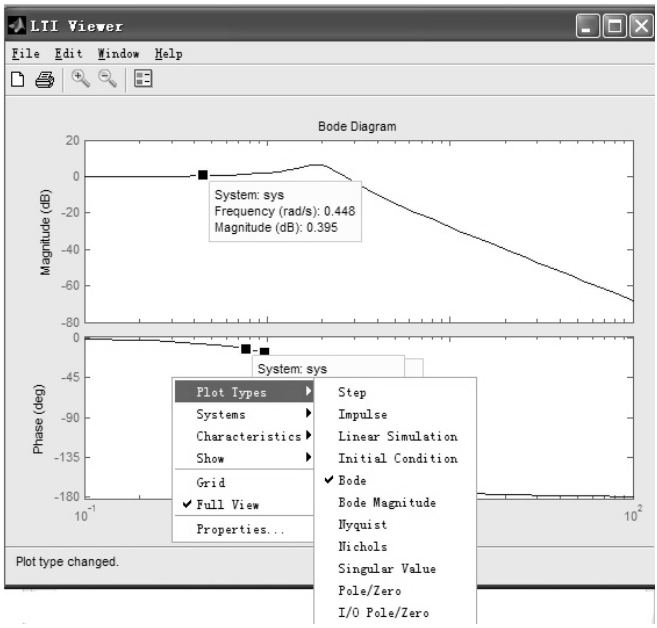


图 7-3-10　ltiview 操作界面

ltiview 也可通过命令方式操作，打开一个系统分析窗口界面，进行线性系统的时域分析或者频域分析。应用格式为

```
ltiview (sys1, …)
ltiview (plottype, sysl, …)
ltiview ('lsim', sysl, sys2, u, t, x0)
```

格式 1：绘制系统对象 sysl，sys2，…的阶跃响应曲线（默认）

格式 2：sysl，…为系统对象；plottype 为分析方法作图，选项如下所述。

- step：阶跃响应分析；
- impulse：脉冲响应分析；
- bode：波特图；
- bodemag：波特图稳定裕度分析；
- nyquist：奈奎斯特图；
- nichols：尼柯尔斯图线；
- sigma：奇异值绘图；
- pzmap：零点、极点绘图。

格式 3：任意输入响应分析。

- u：输入信号向量；
- t：时间向量；
- x0: 初值向量。

【例 7-3-12】使用命令行方式调用 ltiview 工具，作系统时域及频域分析。

程序如下：

```
num=[4];
den=[1 1 4];
sys=tf(num,den);
ltiview({'step';'bode'},sys)
```

系统的阶跃响应图和波特图，如图 7-3-11 所示。

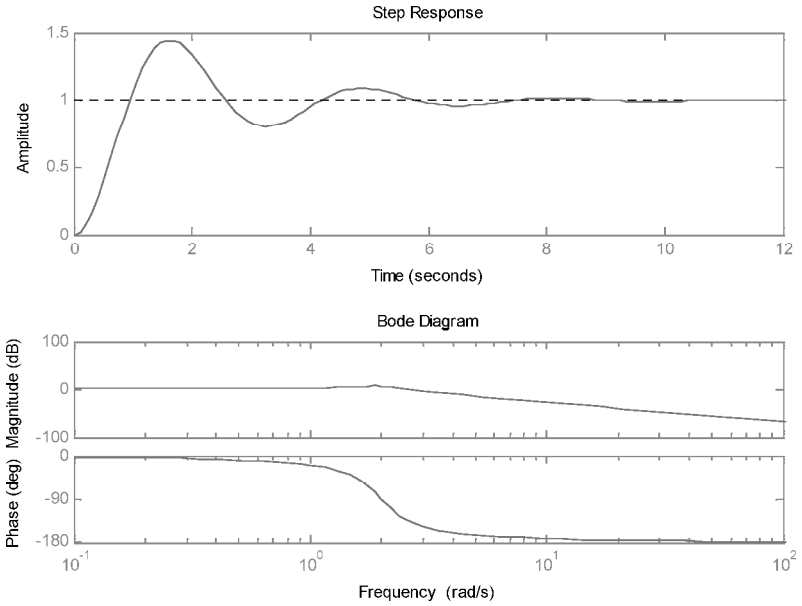


图 7-3-11　系统的阶跃响应和波特图

7.4　控制系统的设计

控制系统的设计有多种方法，MATLAB 中提供了大量控制系统设计的函数，见表 7-4-1。Simulink 中也有用于控制器设计的图形化工具 sisotool。

表 7-4-1　系统设计函数

函 数 格 式	功　　能
acker(A,B,p)	SISO 系统极点配置
place(A,B,p)	MIMO 系统极点配置
estim(sys,L)	生成系统状态估计器
reg(sys,K,L)	生成系统调节器
place(A,B,P)	极点配置
lqr(SYS,Q,R,N)	线性二次调节器设计
dlqr(A,B,Q,R,N	离散线性二次调节器设计
lqry(SYS,Q,R,N)	系统的 LQ 调节器设计
lqrd(A,B,Q,R,Ts)	连续时间系统的离散 LQ 调节器设计

7.4.1 串联校正

在频率特性法中，由开环系统的 Bode 图来分析闭环控制系统稳定性时，通常采用相角裕量和幅值裕量来描述闭环系统的相对稳定性。

串联校正装置的频域设计方法，包括相位超前、相位滞后、相位滞后超前三种校正装置设计。

- 相位超前校正主要用于改善闭环系统的动态特性，对于系统的稳态精度影响较小。
- 相位滞后校正可以明显地改善系统的稳态性能，但会使动态响应过程变缓。
- 相位滞后超前校正则把两者的校正特性结合起来，用于动态、静态特性均要求较高的系统。

【例 7-4-1】给定系统如图 7-4-1 所示，试设计一个串联校正装置，使系统满足幅值裕量大于 10 dB，相位裕量大于等于 45°。

分析：为了满足上述要求，先可以试探地采用超前校正装置 $G_c(s)$，使系统变为如图 7-4-2 的结构。

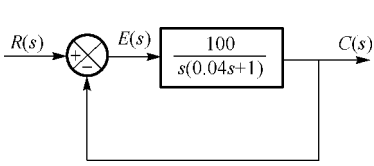


图 7-4-1　校正前系统

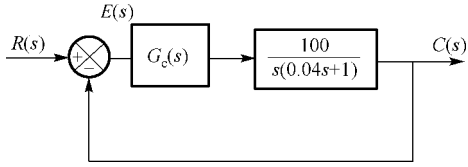


图 7-4-2　校正后系统

先用以下 MATLAB 语句得出原系统的幅值裕量与相位裕量。

```
G=tf(100,[0.04,1,0]);
[Gw,Pw,Wcg,Wcp]=margin(G);
```

在命令窗口中显示如下结果：

```
w =                          Pw =
        Inf                          28.0243
Wcg =                        Wcp =
        Inf                          46.9701
```

可以看出，这个系统有无穷大的幅值裕量，并且其相位裕量$\gamma = 28°$，幅值穿越频率 wcp=47 rad/sec。

引入一个串联超前校正装置 $G_c(s)=\dfrac{0.025s+1}{0.01s+1}$。

可以通过下面的 MATLAB 语句得出校正前后系统的 Bode 图及校正前后系统的阶跃响应图。其中$\omega 1$、$\gamma 1$、ts1 分别为校正前系统的幅值穿越频率、相角裕量、调节时间；$\omega 2$、$\gamma 2$、ts2 分别为校正后系统的幅值穿越频率、相角裕量、调节时间。

程序如下：

```
G1=tf(100,[0.04,1,0]); % 校正前模型
G2=tf(100*[0.025,1],conv([0.04,1,0],[0.01,1])) % 校正后模型
```

```
% 画波特图,校正前用实线，校正后用短画线
bode(G1)
hold
bode(G2,'-- ')
% 画时域响应图，校正前用实线，校正后用短画线
figure
G1_c=feedback(G1,1)
G2_c=feedback(G2,1)
step(G1_c)
hold
step(G2_c,'-- ')
```

校正前后系统的 Bode 图和阶跃响应分别如图 7-4-3 和图 7-4-4 所示。

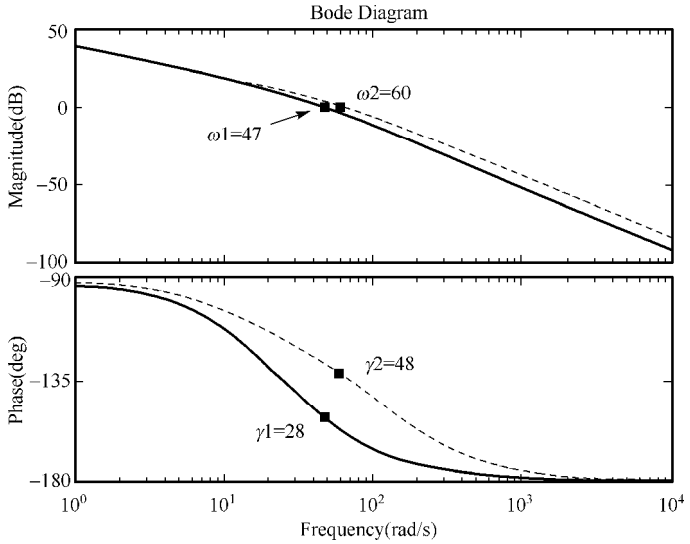


图 7-4-3　校正前后系统的 Bode 图

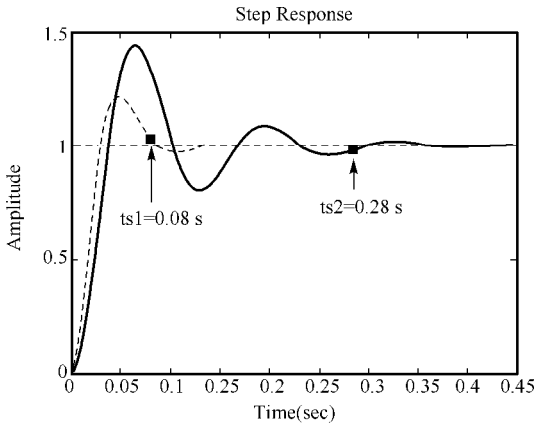


图 7-4-4　校正前后系统的阶跃响应图

可以看出，在这样的控制器下，校正后系统的相位裕量由 28° 增加到 48°，调节时间由 0.28 s 减少到 0.08 s，系统的性能有了明显的提高，满足了设计要求。

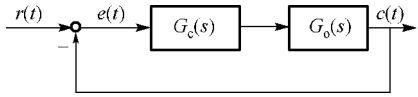
校正后系统的 Bode 图和阶跃响应曲线表明系统具有良好的动态特性，比原来有了明显的改善，校正满足设计要求。

7.4.2 PID 控制器调节

PID 控制器由比例环节、积分环节和微分环节组成，可以实现各种要求的控制规律。

PID 控制器的设计就是确定 PID 控制器的三个参数，即比例系数 P_k、积分系数 I_k（积分时间常数 T_i）和微分系数 K_d（微分时间常数 T_d）。在 MATLAB 环境中，PID 参数的选择通常采用 Ziegler-Nichols 经验整定方法。

【例 7-4-2】已知控制系统框图如下所示。



其中，被控对象 $G_o(s)=\dfrac{10}{0.8s+1}e^{-0.1s}$，$G_c(s)$为控制器，试建立 Simulink 仿真模型，并利用 Ziegler-Nichols 法整定 PID 控制器参数，要求调整到系统为无差，超调量为 8%。

分析：(1)建立加上控制器之前的系统 Simulink 模型，如图 7-4-5 所示。其中 Transport-Delay 模块中“Time Delay”参数设为 0.1，设置仿真时间为 4 s，其单位阶跃响应如图 7-4-6 所示。由图可得，其超调量达到了 45%，稳态误差为 0.1，一般不符合要求。

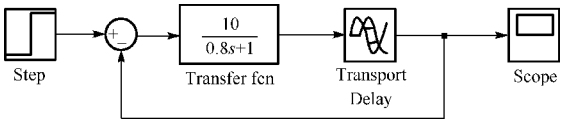


图 7-4-5　系统仿真模型

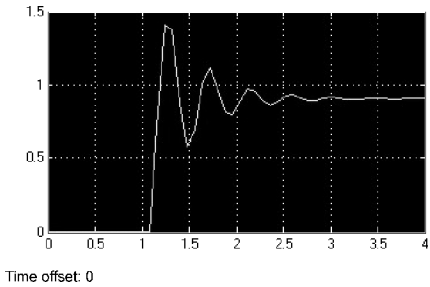


图 7-4-6　校正前阶跃响应

(2)建立加上 PID 校正装置的 Simulink 模型，如图 7-4-7 所示。设初始参数 $K_p=\dfrac{1.2T}{K\sigma}=0.96$，$T_I=2\sigma=0.2$，$T_D=0.5\sigma=0.05$，观察仿真结果。经过多次调整，$K_p=0.3$，$T_I=0.6$，$T_d=0.05$，得到如图 7-4-8 所示阶跃响应。可见此时超调量已经非常小，调整时间也小于 2 s，稳态没有误差。

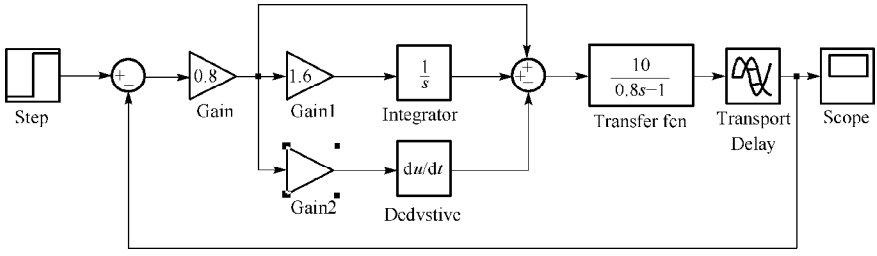


图 7-4-7　加入 PID 控制的仿真模型

读者可自行修改 PID 的参数，观察比例、积分、微分各控制环节参数的修改对控制效果的影响。

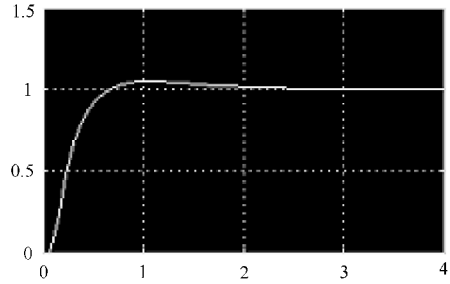


图 7-4-8　校正后的系统响应曲线

7.4.3　状态反馈与极点配置

状态反馈是将系统的状态变量乘以相应的反馈系数，然后反馈到输入端与参考输入叠加形成控制，作为受控系统的控制输入。采用状态反馈不但可以实现闭环系统极点的任意配置，而且也是实现解调和构成线性最优调节器的主要手段。

1．极点配置

对于线性定常系统，系统的稳定性和各种性能的品质指标，在很大程度上是由闭环系统的极点位置所决定的。因此在进行系统设计时，设法使闭环系统的极点位于 s 平面上的一组合理的、具有所期望的性能品质指标的极点，可以有效地改善系统的性能品质指标。

在经典控制理论的系统综合中，无论采用频率域法还是根轨迹法，都是通过改变极点的位置来改善性能指标的，本质上均属于极点配置方法。

线性系统的动态性能，如系统稳定性、时间域分析中的超调量、过渡时间等指标，主要取决于系统的极点位置。极点配置的一般方法可以通过换算（如根轨迹法）和经验估计来具体地加以确定。把闭环极点组配置到所希望的位置上，等价于使综合得到的线性系统的动态性能达到期望的要求。线性定常系统可通过状态变量反馈来任意配置其全部极点的充要条件是该系统为完全可控的。因此，对于一个完全可控的线性系统的极点配置问题，实际上转化为求解状态反馈增益矩阵 $\boldsymbol{K}$。

2．状态观测器的设计

极点配置是基于状态反馈的，因此状态 X 必须可观测。当状态不能观测时，则应设计状态观测器来估计状态。包括全维观测器和降维观测器的设计。

状态观测器的设计思路是：采用被测系统的可测参量，如输出量 y 和输入量 u，重新构造状态 $\tilde{x}$，使状态反馈得以实现。

【例 7-4-3】已知控制系统的状态方程为

$$\boldsymbol{A}=\begin{bmatrix}-10 & -35 & -50 & -24\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\qquad \boldsymbol{B}=\begin{bmatrix}1\\0\\0\\0\end{bmatrix}\qquad \boldsymbol{C}=[1\ \ 7\ \ 24\ \ 24]$$

判断系统的稳定性，绘出单位阶跃响应曲线；对系统进行可控性、可观测性分析，以及极点配置控制器的设计。要求闭环极点配置在 $p = [-30, -1.2; -2.4 \pm 4i]$ 位置上。

程序如下：

（1）判断系统的稳定性程序如下所示，响应曲线如图 7-4-9 所示。

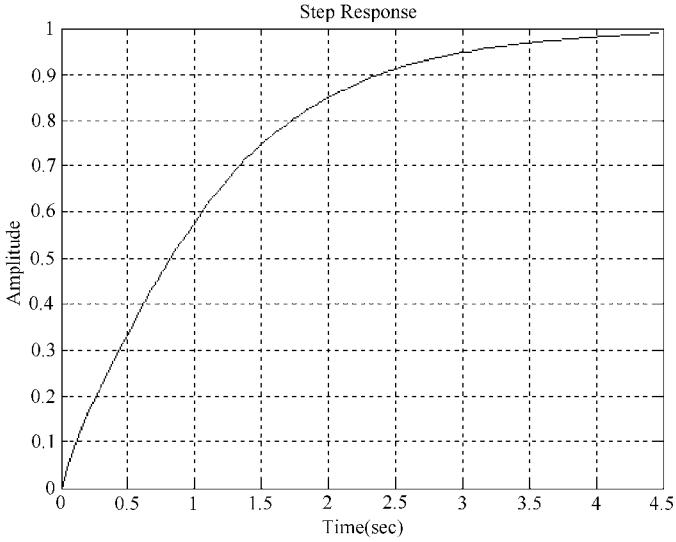


图 7-4-9　系统阶跃响应曲线

```
A=[-10 -35 -50 -24;1 0 0 0; 0 1 0 0;0 0 1 0];
B=[1;0;0;0];
C=[1 7 24 24];
D=[0];
[z,p,k]=ss2zp(A,B,C,D);
flagz=0;
n=length(A);
for i=1:n
    if  real(p(i))>0
        flagz=1;
    end
end
disp('系统的零极点模型:');z,p,k
if flagz==1
    disp('系统不稳定.');
else
    disp('系统是稳定的.');
end
step(A,B,C,D);grid
```

运行结果为

```
系统的零极点模型:
z=
-2.7306+2.8531i
-2.7306-2.8531i
-1.5388
p=
-4.0000
-3.0000
-2.0000
-1.0000
k=
1.0000
系统是稳定的。
```

（2）判断系统的可控性，求解系统的变换矩阵 $\boldsymbol{Q}$。如果系统完全可控，导出系统的第一可控标准型。

```
Q=ctrb(A,B);
m=rank(Q);
if m==n
Ac=inv(Q)*A*Q;Bc=inv(Q)*B;Cc=C*Q;
disp('系统是可控的');
disp('系统第一可控标准型');Ac,Bc,Cc
disp('变换矩阵');Q
else
disp('系统状态不完全可控');
disp('可控的状态变量数');m
end
```

运行结果为：

```
系统是可控的
系统第一可控标准型
Ac=
0     0     0   -24
1     0     0   -50
0     1     0   -35
0     0     1   -10
Bc=
1
0
0
0
Cc=
1    -3    19  -111
```

```
变换矩阵
Q=
1  -10    65 -350
0    1  -10    65
0    0    1  -10
0    0    0    1
```

（3）判断系统的可观测性，求解系统的变换矩阵 $\boldsymbol{P}$。如果系统完全可观测，导出系统的第一可观测标准型。

```
Qo=obsv(A,C);
mm=rank(Qo);
if mm==n
P=inv(Qo);Ao=inv(P)*A*P;Bo=inv(P)*B;Co=C*P;
disp('系统是可观测的.');
disp('系统第一可观测标准型:');Ao,Bo,Co
disp('变换矩阵:');P
else
disp('系统状态不完全可观.');
disp('可观测的状态变量数为:');mm
end
```

运行结果为：

```
系统是可观测的.
系统第一可观测标准型:
Ao=
-0.0000    1.0000  -0.0000  -0.0000
0.0000    0.0000    1.0000    0.0000
-0.0000    0.0000    0.0000    1.0000
-24.0000 -50.0000 -35.0000 -10.0000
Bo=
1.0000
-3.0000
19.0000
-111.0000
Co=
1.0000 0  -0.0000  -0.0000
变换矩阵:
P=
-7.3333  -7.8611  -3.0417  -0.3472
4.3333    7.1944    2.2917    0.2639
-2.8333  -4.0278  -1.5417  -0.1806
1.8750    2.5903    1.0000    0.1181
```

（4）确定状态反馈矩阵 $\boldsymbol{K}$，使系统的闭环极点配置在 p = [–30, –1.2; –2.4±4i]位置上，并绘制出配置后的单位阶跃响应曲线。MATLAB 利用函数 place（A,B,P）可求得状态反馈矩阵 $\boldsymbol{K}$。

```
disp('原系统的极点:');p=eig(A)'
P=[-30;-1.2;-2.4+sqrt(-16);-2.4-sqrt(-16)];
disp('状态反馈增益:');
K=place(A,B,P)
disp('配置后系统的极点:');p=eig(A-B*K)'
disp('极点配置后的闭环系统:')
sysnew=ss(A-B*K,B,C,D)
figure(2)
step(sysnew/dcgain(sysnew));
grid
```

运行结果为：

```
原系统的极点：
p=
-4.0000 -3.0000 -2.0000 -1.0000
状态反馈增益：
K=
27.0000  172.5200  801.7120  759.3600
配置后系统的极点：
p=
-30.0000   -2.4000-4.0000i   -2.4000+4.0000i    -1.2000
极点配置后的闭环系统：
a=
x1      x2      x3      x4
x1     -36   -207.5   -851.7   -783.4
x2      1       00      0
x3     0     1      0     0
x4     0     0      1     0
b=
u1
x1   1
x2   0
x3   0
x4   0
c=
x1  x2  x3  x4
y1   1   7  24  24
d=
u1
y1   0
Continuous-time model.
```

极点配置后系统的阶跃响应曲线如图 7-4-10 所示。

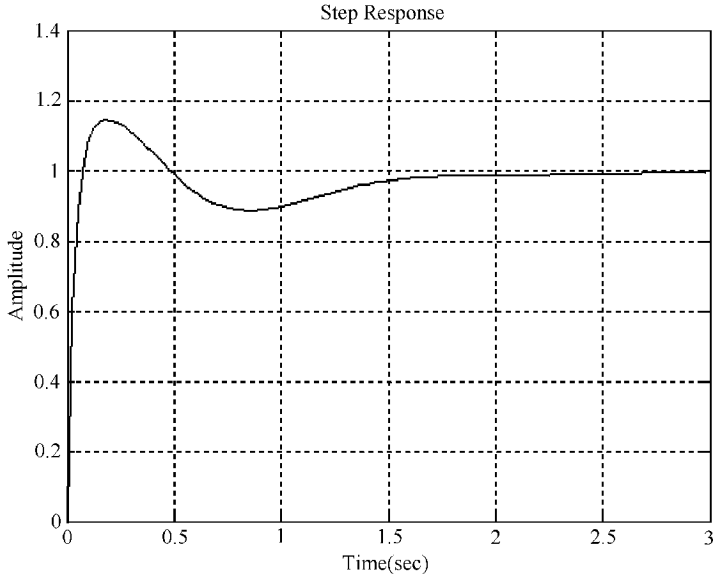


图 7-4-10　极点配置后系统阶跃响应曲线

7.4.4　系统设计工具 SISO Design Tool

SISO Design Tool 为单输入单输出系统的 GUI 设计工具，它可完成的功能有：

- 利用根轨迹方法计算系统闭环特性；
- 针对开环系统 Bode 图进行系统设计；
- 添加补偿器的零极点；
- 设计超前、滞后网络和滤波器；
- 分析闭环系统响应；
- 调整系统幅值或相位裕度；
- 系统在连续和离散状态下的互换。

类似于 ltiview，sisotool 也可用菜单和命令两种方式进行操作。

命令格式如下：

```
sisotool(G)
sisotool(G,C)
sisotool(G,C,H,F)
sisotool(view,G,…)
```

格式 1：给定线性系统的开环对象模型 G 可以对 tf、zpk、ss 对象任意一种作系统分析。

格式 2：增加补偿器 C 作系统分析。

格式 3：对定系统的各环节 G、C、H、F 作系统分析，其中

- G：系统对象模型；
- C：补偿器对象模型；
- H：反馈对象模型；

- F：前置滤波器模型。

格式 4：指定 view 的作图类型，view 可以为：

- rlocus，根轨迹图；
- bode，波特图；
- nichols，尼柯尔斯图线；
- filter，前置滤波器的 Bode 图。

【例 7-4-4】系统开环传递函数为

$$G(s)=\frac{7(s+1)}{s(s+3)(s^2+4s+5)}$$

使用 sisotool 工具，并作系统分析。

```
num1=[7 7];     %计算 1-φ(s)的值.
den1=[conv(conv([1 0],[1 3]),[1 4 5])];
G=tf(num1,den1);
GG=feedback(G,1,-1);
sisotool(GG)
sisotool({bode,nichols},sys)
```

系统的零极点配置如图 7-4-11 所示，系统的 Bode 图和 Nichols 图如图 7-4-12 所示。

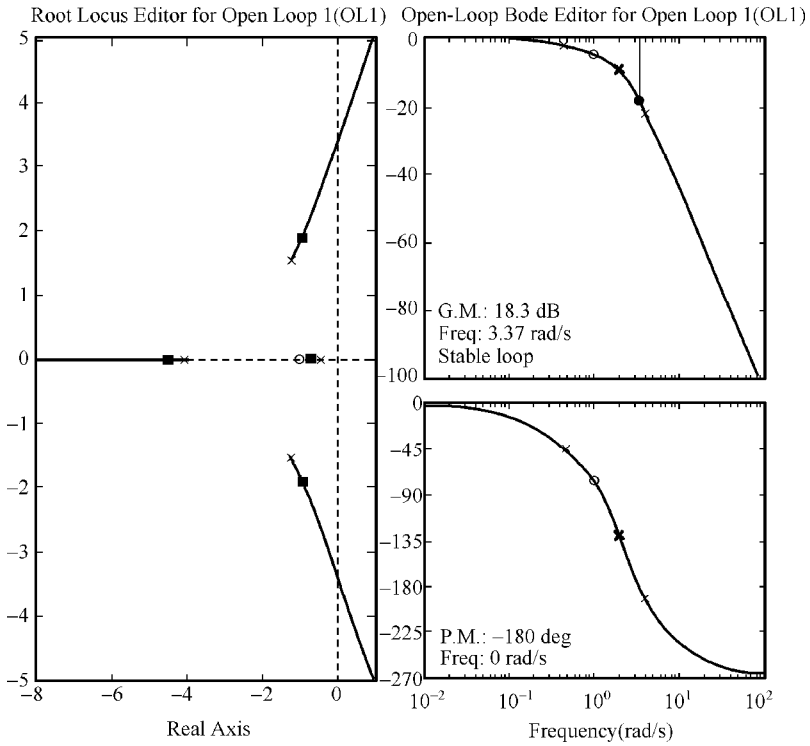


图 7-4-11　零极点配置图

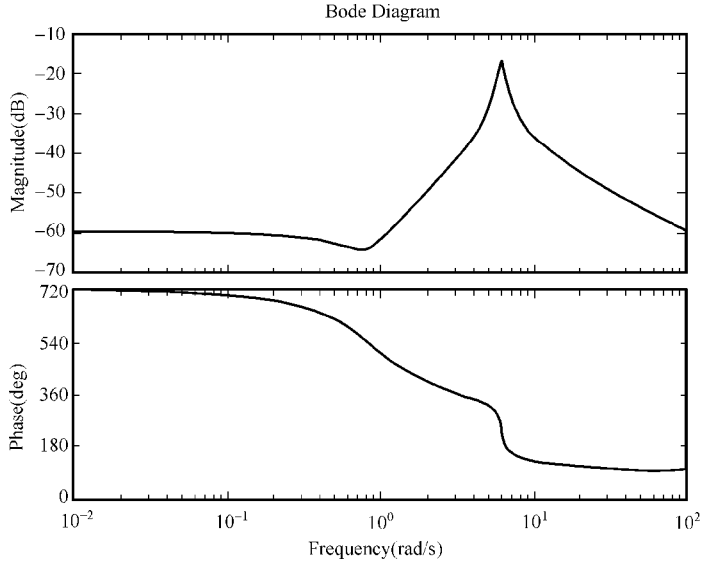


图 7-4-12　Bode 图和 Nichols 图

7.5　仿真实例——直流电机双闭环调速

直流电机调速系统因其调速范围广、静差率小、稳定性好以及具有良好的动态性能等特性，在高性能的拖动技术领域中得到了广泛的应用。双闭环直流调速系统是直流调速控制系统中发展最为成熟，应用非常广泛的电力传动系统。

7.5.1　系统的工作原理

双闭环控制直流调速系统的特点是电动机的转速和电流分别由两个独立的调节器控制，且转速调节器的输出就是电流调节器的给定，因此电流环能跟随转速的偏差调节电动机电枢电流。当转速低于给定转速时，转速调节器的积分作用使输出增加，即电流给定上升，并通过电流环调节使电动机电流增加，从而使电动机获得加速转矩，电动机转速上升。当实际转速高于给定转速时，转速调节器的输出减小，即电流给定减小，并通过电流环调节使电动机电流下降，电动机将因为电磁转矩减小而减速。当转速调节器饱和输出达到限幅时，电流环即以最大电流限制，实现电动机的加速，使电动机的启动时间最短，其原理如图 7-5-1 所示。

双闭环调速系统的两个调节器 ASR 和 ACR 一般都采用 PI 调节器，因为 PI 调节器作为校正装置既可以保证系统的稳态精度，使系统在稳态运行时得到无静差调速，又能提高系统的稳定性；作为控制器时又能兼顾快速响应和消除静差两方面的要求。

1．转速调节器的作用

（1）使转速 n 跟随给定电压变化，当偏差电压为零时，实现稳态无静差。

（2）对负载变化起抗扰作用。

（3）其输出限幅值决定允许的最大电流。

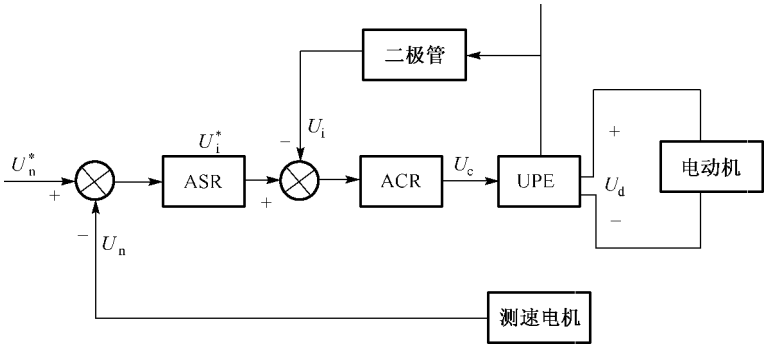


图 7-5-1　转速、电流双闭环直流调速系统原理图

2．电流调节器的作用

（1）在转速调节过程中，使电流跟随其给定电压变化。

（2）对电网电压波动起及时抗扰作用。

（3）起动时保证获得允许的最大电流，使系统获得最大加速度起动。

（4）当电机过载甚至于堵转时，限制电枢电流的最大值，从而起到快速的安全保护作用；当故障消失时，系统能够自动恢复正常。

7.5.2　系统的动态性能分析

一般来说，双闭环调速系统具有比较满意的动态性能。动态性能可分为动态跟随性能和动态抗扰性能两种，其中动态抗扰性能对于调速系统更为重要，它主要表现为抗负载扰动和抗电网电压扰动。

1．动态跟随性能

双闭环调速系统在起动和升速过程中，能够在电流受电机过载能力约束的条件下，表现出很快的动态跟随性能。在减速过程中，由于主电路电流的不可逆性，跟随性能变差。在设计 ACR 时，应强调具有良好的跟随性能。

2．动态抗扰性能

1）抗负载扰动。

负载扰动作用在电流环之后，因此只能靠转速调节 ASR 来产生抗负载扰动的作用。在负载突变时，必然会引起动态指标突变。为了减少动态指标突变，要求 ASR 具有较好的抗扰性能。

2）抗电网电压扰动。

由于电网电压扰动和负载扰动在系统结构图中作用的位置不同，系统对它们的动态抗扰效果就不同。电网电压扰动和负载扰动都作用在被转速负反馈环包围的前向通道上，就静特性而言，系统对它们的抗扰效果是一样的。从动态性能上看，负载扰动作用在被调量的前面，可以通过测速发电机检测出来，使负载扰动通过转速负反馈得到及时调节。而电网电压扰动作用在离被调量更远的位置，转速调节器 ASR 不能及时对它进行调节，但是因

为它作用在被电流负反馈环包围的前向通道上，使电压波动可以直接通过电流反馈得到及时的调节，不必等它影响到转速以后才能反馈回来，抗扰性能高。在双闭环系统中，由电网电压波动引起的转速动态变化比起单闭环系统小得多。

7.5.3 系统的数学模型和仿真模型

双闭环控制系统数学模型的主要形式仍然是以传递函数或零极点模型为基础的系统动态结构图，双闭环直流调速系统的动态结构框图如图 7-5-2 所示，图中 $W_{ASR}(s)$ 和 $W_{ACR}(s)$ 分别表示转速调节器和电流调节器的传递函数。

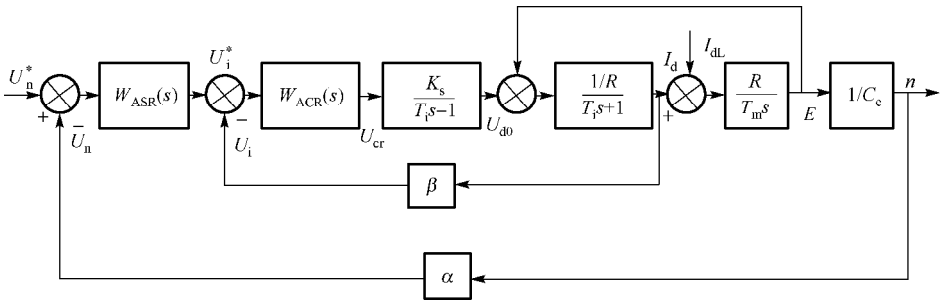


图 7-5-2 双闭环直流调速系统的动态结构框图

依照结构框图，在 Simulink 选取相应的模块可构建晶闸管-直流电机双闭环调速系统的 Simulink 动态结构图，如图 7-5-3 所示，设置直流电机参数，忽略系统的非线性，分别对系统的电流内环与转速外环进行稳态与动态的计算及仿真。

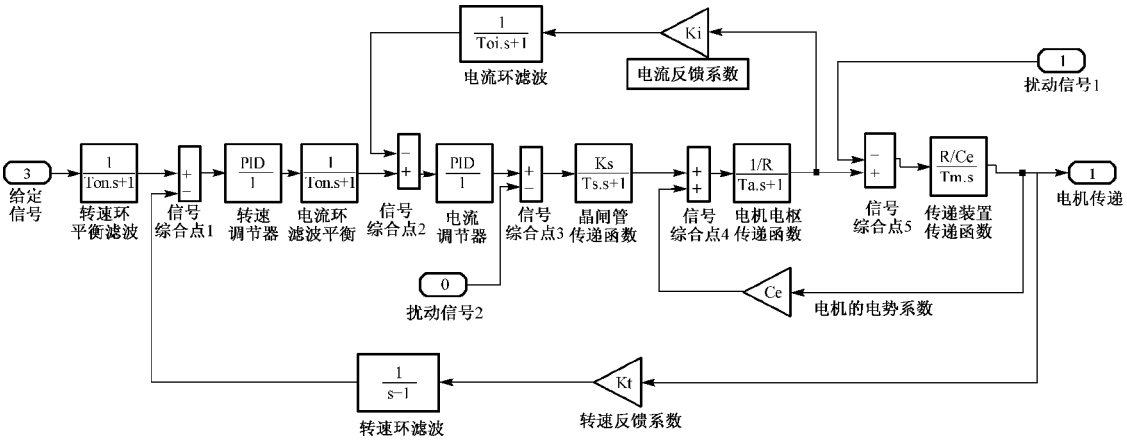


图 7-5-3 双闭环调速系统的 Simulink 动态结构图

7.5.4 调节器的设计

在双闭环直流调速系统中，电流环的主要作用就是保证电机起动时获得允许的最大电流，保持电枢电流在动态过程中不超过允许值，突加控制作用时超调量越小越好。根据自动控制理论，将电流环设计成典型 I 型系统，因为典型 I 型系统动态跟随性能的超调量很小，符合电流环的设计要求；而转速环在系统中主要发挥其抗扰动作用，根据自动控制理

论，将转速环设计成典型 II 型系统，因为典型 II 型系统动态抗扰性能的动态速降小，符合转速环的设计要求。

按照把电流环设计成 I 型系统的要求，根据自动控制理论，电流调节器应选择比例积分调节器，即 PI 调节器，其传递函数为

$$W_{\mathrm{ACR}}(s)=K_{\mathrm{pi}}\frac{T_{\mathrm{i}}s+1}{T_{\mathrm{i}}s}=\frac{0.0128s+1}{0.04s},\qquad K_{\mathrm{pi}}=0.32;\qquad T_{\mathrm{i}}=0.0128=T_{\mathrm{a}}$$

按照把转速环设计成 II 型系统的要求，根据自动控制理论，转速调节器也应选择比例积分调节器，其传递函数为

$$W_{\mathrm{ASR}}(s)=K_{\mathrm{pn}}\frac{T_{\mathrm{n}}s+1}{T_{\mathrm{n}}s},\qquad T_{\mathrm{n}}=0.0867$$

为保证电流调节器和转速调节器的运算放大器工作在线性特性段以及保护调速系统的各个元件、部件与装置不致损坏，在电流调节器和转速调节器的输出端都设置了限幅装置。

因为电流检测信号中常含有交流分量，所以须添加低通滤波器，但是由低通滤波器产生的反馈滤波同时也给反馈信号带来延滞，为平衡这一延滞作用，在给定信号通道也添加一个与反馈滤波相同时间数的惯性环节，使得给定信号与反馈信号经过同样的延滞。其传递函数为

$$\frac{1}{T_{\mathrm{oi}}s+1}$$

与电流环添加低通滤波器措施一样，在转速环反馈通道与给定信号通道都添加了滤波惯性环节，其传递函数与上述相同。

7.5.5 调节器校正前后的典型响应分析

1．电流环时域分析

电流环的校正主要是对晶闸管整流与移相触发装置的放大倍数 K_{s} 进行校正，假设校正前 $K_{\mathrm{s}}=20$，构成动态结构图模型 mx01.mdl，如图 7-5-4 所示；校正后 $K_{\mathrm{s}}=30$，构成动态结构图模型 mx02.mdl。

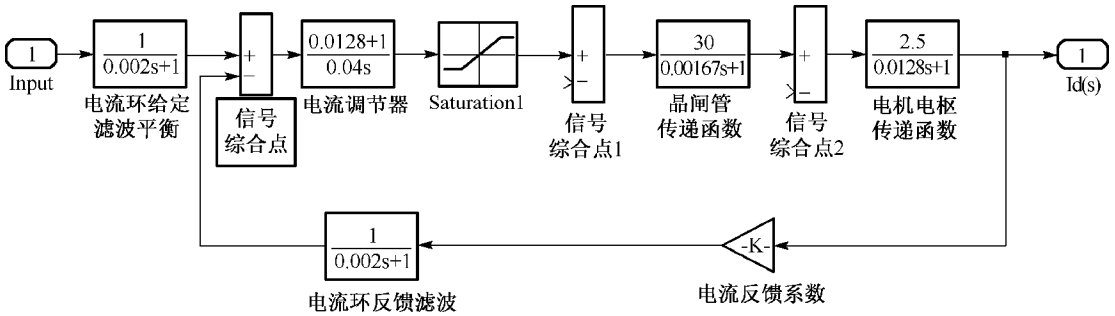
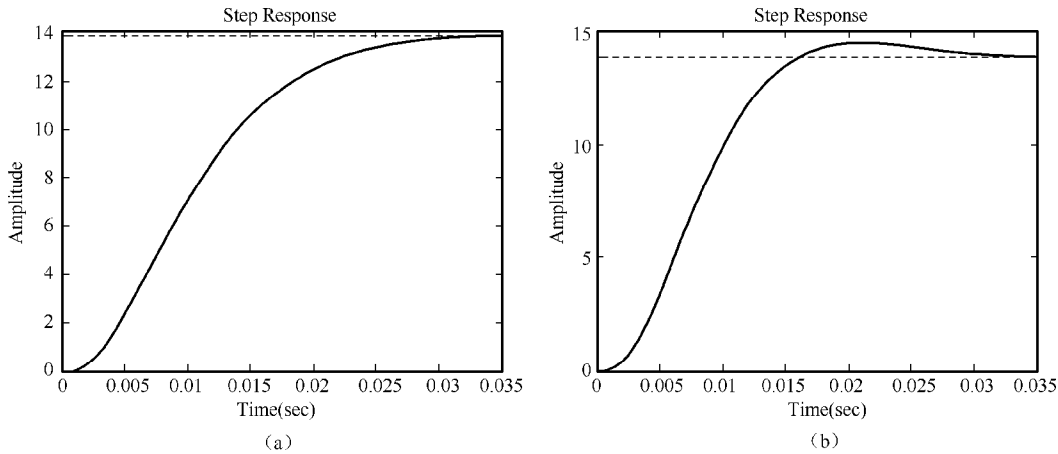


图 7-5-4　带参数电流环的 Simulink 模型 mx01.mdl

执行以下程序：

```
[a1,b1,c1,d1]=linmod('mx01');s1=ss(a1,b1,c1,d1);
figure(1);step(s1);hold on
[a2,b2,c2,d2]=linmod('mx02');s2=ss(a2,b2,c2,d2);
figure(2);step(s2)
[y,t]=step(s1);[mp,tf]=max(y);cs=length(t);
yss=y(cs);sgm=100*(mp-yss)/yss
tp=t(tf)
```

运行该程序可得模型 mx01.mdl 与 mx02.mdl 的单位阶跃响应曲线，如图 7-5-5 的（a）与（b）所示，并对于图 7-5-5 的（b）图求出性能指标，超调量$\sigma\%=4.4403\%$，峰值时间$t_p=0.0209$ s。



（a）　　（b）

图 7-5-5　电流环阶跃响应 Simulink 曲线

图 7-5-5（a）是$K_s=20$ 时的系统单位阶跃响应，阶跃响应曲线单调上升，完全无超调，并且在 0.04 s 内响应即结束。这样的电流环阶跃响应很理想，但是电机的加速起动不够快。

图 7-5-5（b）是$K_s=30$ 时的系统单位阶跃响应，响应曲线略有超调 4.4403%，符合 I 型系统超调量小的特点，系统曲线迅速上升，峰值时间非常短，电流立即下降至恒定并在 0.04 s 内响应即结束，这样的阶跃响应是很理想的。对于电流环，比较此二者，电流稍微超调的可取，因为这有利于电机的加速起动，电机又不受什么影响。

2．电流环频域分析

根据自动控制原理，频域分析的特点是运用闭环系统的开环频率特性曲线来分析闭环系统的响应及其性能。频域分析的主要内容是画 Bode 图与计算频域性能指标。电流闭环系统的开环结构图如图 7-5-6 所示，它对应着 Simulink 动态结构图模型 mx03.mdl.

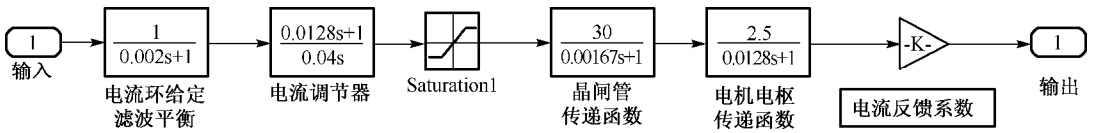


图 7-5-6　电流闭环系统的开环结构图 mx03.mdl

执行以下程序：

```
n1=1;d1=[0.002 1];s1=tf(n1,d1);
n2=[0.0128 1];d2=[0.04 0];s2=tf(n2,d2);
n3=30;d3=[0.00167 1];s3=tf(n3,d3);
n4=2.5;d4=[0.0128 1];s4=tf(n4,d4);
n5=0.072;d5=[1];s5=tf(n5,d5);
sys=s1*s2*s3*s4*s5;
margin(sys)
```

执行语句后，可得电流环的 Bode 图，如图 7-5-7 所示，在图上附有经计算出电流环的频域性能指标。

模稳定裕度 L_h=18.2 dB，–π穿越频率 W_g = 547 rad/s。

相稳定裕度γ=63.6°，剪切频率 W_c= 128 rad/s。

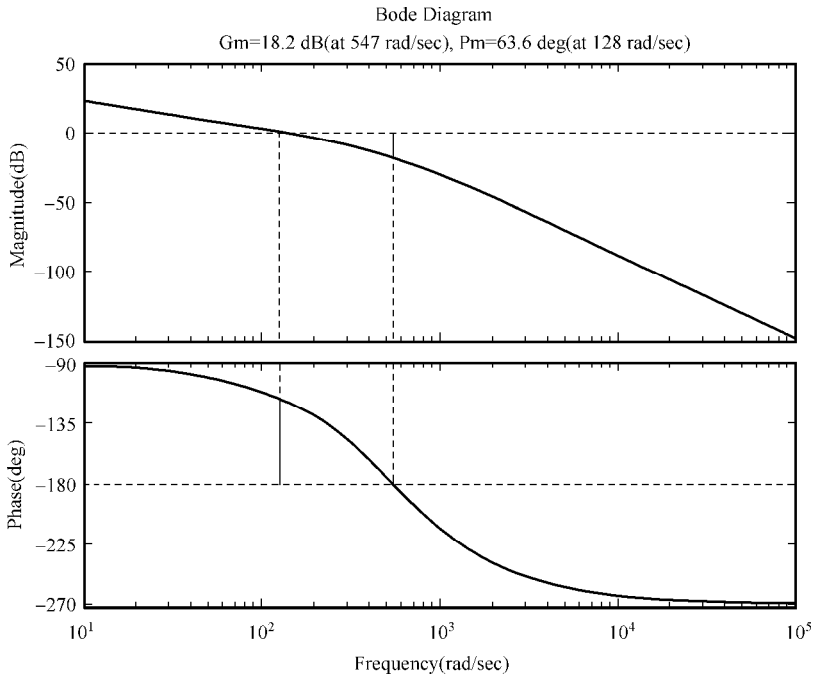


图 7-5-7　电流环的 Bode 图

工程上，一般要求模稳定裕度大于等于 6 dB，相稳定裕度大于等于 40°，可见电流环有足够的稳定裕量，其频域性能是优良的，反映了电流环具有良好的相对稳定性。

转速环的校正主要是对转速调节器的参数 K_{pn} 以及对晶闸管整流与移相触发装置的放大倍数 K_s 进行校正。读者可参考电流环的校正进行编程。

第 8 章　电力系统仿真

电力系统是集发电、输电、配电和用电为一体的复杂非线性网络系统，是一种多调节量、多目标、非线性、变参数的复杂系统，对其物理本质的研究涉及到短至 1 μs 到长至 1 h 的动态过程。不像线性系统那样，系统调试可以逐个依次调整，使系统处于稳定和最佳状态。必须以数学建模、计算机仿真为基础，达到整体优化，以此来确定各调整点的最佳整定值。

8.1　电力系统仿真概述

电力系统一般由发电机、变压器、电力线路及电力负荷组成。电力电子技术综合了电子电路、电机拖动、计算机控制等多学科知识，由于电力电子器件自身的开关非线性，给电力电子电路的分析带来了一定的复杂性和困难，仿真的价值在于可以不用考虑实际实现的巨大成本，先用计算机验证所设计的电路的正确性、功能及性能，以一种直观的形式增加对基本电路的认知，可以随意的改变某一参数观察其对电路性能的影响，随意改变某些连线观察结果，可以随意制造故障等。仿真方法包括离线数字仿真和实时数字仿真。

1．离线数字仿真

电力系统离线数字仿真通过建立电力系统物理过程的数学模型，用求解数学方程的方法来进行仿真研究。目前主要利用电力系统仿真软件离线计算的方法对电力系统及装置的动态行为进行仿真研究。

根据动态过程中系统模型和仿真方法的不同，离线数字仿真可以分为电磁暂态过程仿真、机电暂态过程仿真和中长期动态过程仿真。

电磁暂态数字仿真是用数值计算方法对电力系统中从数微秒至数秒之间的电磁暂态过程进行仿真模拟。

机电暂态数字仿真主要研究电力系统受到大扰动后的暂态稳定和受到小扰动后的静态稳定性能。

电力系统中长期动态过程仿真是电力系统受到扰动后较长过程的动态仿真，主要用来分析电力系统内较长时间的动态特性。

2．实时数字仿真

电力系统实时数字仿真系统是基于现代计算机技术开发的体系机构和大型电力系统电磁暂态仿真软件系统，可以进行电力系统电磁暂态的全过程实时模拟，不仅是研究电力系统规划、设计和运行的强有力工具，而且通过高速通信系统及信号放大、转换系统与实际

的保护和控制装置相连，对电网中继电保护装置、安全自动装置及一些测控装置进行实时测试，也能对电网中的各种故障和稳定特性进行深入分析，提供技术支持，对系统网络的安全稳定性进行评估。

电力系统实时数字仿真必须按照实际系统运行的时序要求来完成仿真过程的每一个步骤，要求仿真模型的时间比例尺完全等于原始模型的时间比例尺。

与非实时数字仿真相比，实时数字仿真算法有以下几个要求：

- 算法的快速性；
- 算法执行中数据的可取性，即算法所用的信息与实时输入是一致的；
- 算法的鲁棒性；
- 算法的相容性。

目前在电力系统领域的仿真软件大体上分为以下两类：一类是通用电路仿真软件，如MATLAB、SPICE、SABER，通过对其原有的小功率器件模型加以改造，引入功率器件模型，使其应用领域扩展到电力电子器件的仿真。二是专用电力系统仿真软件，如机电暂态仿真工具 PSS/E、PSASP、BPA，电磁暂态仿真工具 EMTP/ATP、PSCAD/EMTDC、EmtpE，电磁场仿真工具 ANSOFT、ANSYS，电力电子仿真工具 PLECS、PSIM，配电网仿真工具 ETAP、EDSA/DesignBase 等。

8.2 Simpowersystems 模块库

8.2.1 Simpowersystems 简介

Simpowersystems 是在 Simulink 环境下进行电力电子系统建模和仿真的模块库，该模块库包含了各种交/直流电源、电气网络中常见的元器件和电工测量仪表以及分析工具等，以直观易用的图形方式对电气系统进行模型描述。利用这些模块可以模拟电力系统运行和故障的各种状态，并进行仿真和分析。模型还可与其他 Simulink 模块相连接，进行一体化的系统级动态分析。

Simpowersystems 模块中的数学模型基于成熟的电磁和机电方程，用标准的电气符号表示，它在发电输变电系统和电力分配计算方面提供了强有力的解决方法，尤其是当设计开发内容涉及控制系统设计时，优势更为突出。对于具有复杂自给型供电装置，如汽车、飞机、制造设备上的电气系统和普通用电装置而言，用 Simpowersystems 进行分析和设计也非常理想。

1．仿真模型

Simpowersystems 提供了两种电力系统连续仿真：离散化仿真和矢量化仿真。

离散化仿真可以将模型离散化，使用定步长梯形积分法进行离散仿真计算。这一特性能够显著提高仿真计算的实时性，尤其是那些带电力电子设备的模型。另外，由于模型被离散化，还可以用 Real-TimeWorkshop 生成模型的代码，进一步提高仿真的速度。

矢量化仿真利用一些的固有频率的代数方程取代描述电子网络的微分方程，矢量化仿真可改善机械系统瞬态研究的稳定性。

2．求解技术

Simpowersystems 与 Simulink 和 MATLAB 是无缝结合的。使用 Simulink 的求解技术可以提供快速精确的仿真结果。求解的过程同时支持代数约束和状态描述，电力系统仿真中涉及很多开关切换，需要了解系统的瞬间变化性能。Simulink 中一些变步长的求解技术是专门用来针对刚性系统进行建模仿真的，Simulink 的过零检测功能确保了能够以足够的机械精度来检测和处理系统中的中断过程。

Simpowersystems 以 M-文件形式提供了 power2sys 函数，可用于在仿真过程之外获得电路的状态空间模型表达，该函数分析电网络拓扑结构，并计算出等价状态空间模型。在这个函数所提供信息的帮助下，可以使用诸如控制系统工具箱进行更进一步的分析。

3．参数设定和仿真分析

Simpowersystems 提供了 Powergui 模块，用户可以修改模型的初始状态，从任何起始条件开始仿真分析。在交互式窗口上可以显示稳态电压和电流；显示并修改初始状态量；计算潮流和初始化机电模块；当模型中存在电抗测量模块时可显示电抗相对频率的变化；可使用控制系统工具箱的 LTI Viewer 工具，进行系统的时域、频域响应分析；生成稳态计算分析报告。

4．测量显示

Simpowersystems 的测量模块可以将线路中所测量的信号，如电压、电流值等转变为 Simulink 模型信号，并在示波器中显示。电动机和电力电子模块的测量输出端也可以直接输出 Simulink 模型信号。

8.2.2 模块库

Simpowersystems 中含有 100 多个模块，分布在 8 个子库中，这 8 个子库分别如下所述。

1．Electrical Sources（电源子库）

电源子库提供了各种电源，如交流电压源、交流电流源、直流电压源、受控电压源、受控电流源、三相电源、三相可编程电压源模块等，模块说明见表 8-2-1。

表 8-2-1　电源子库

模　块　名	用　　途
AC Current Source	正弦交流电流源
AC Voltage Source	正弦交流电压源
Controlled Current Source	输出电流受输入信号控制的可控电流源
Controlled Voltage Source	输出电压受输入信号控制的可控电压源
DC Voltage Source	直流电压源
3-Phase Programmable Voltage Source	三相可调节电源信号，其中幅值、相角、频率和谐波均可变
3-Phase Source	带有电阻和电感的三相电压源

2．Elements（元件子库）

元件子库提供常用的电器元件模块，如 RLC 支路和负载、线性和饱和变压器、电路分离器、传输线模型等，模块说明见表 8-2-2。

表 8-2-2　元件子库

模　块　名	用　　途
Breaker	断路器（模拟空气开关等）
Connection Port	物理接口端子
Distributed Parameters Line	分布参数线路模块
Ground	接地
Linear Transformer	三绕组线性变压器（单相）
Multi-Winding Transformer	多绕组变压器
Mutual Inductance	三项耦合线圈
Neutral	中性点
Parallel RLC Branch	并联 RLC 支路
Parallel RLC Load	并联 RLC 负荷
PI Section Line	分布电容、电感为 PI 型的传输导线
Saturable Transformer	饱和变压器
Series RLC Branch	串联 RLC 支路
Series RLC Load	串联 RLC 负荷
Surge Arrester	避雷针
3-Phase Breaker	三相断路器
3-Phase Dynamic Load	有功功率和无功功率可调节的三相三绕组动态负荷
3-Phase Fault	三项可变故障断路器
3-Phase Harmonic Filter	三相谐波滤波器
3-Phase Mutual Inductance Z1-Z0	用正序和零序参数表示的三项耦合电感
3-Phase Parallel RLC Branch	三相并联 RLC 支路
3-Phase Parallel RLC Load	三相并联 RLC 负荷
3-Phase PI Section Line	三相 PI 型电路
3-Phase Series RLC Branch	三相串联 RLC 支路
3-Phase Series RLC Load	三相串联 RLC 负荷
Three-Phase Transformer(Three Windings)	三相三绕组变压器
Three-Phase Transformer(Two Windings)	三相双绕组变压器
3-Phase Transformer 12-terminals	三个单项双绕组变压器组成的模块，所有端口可见
Zigzag Phase-Shifting Transformer	Z 形移相变压器

3．Machines（电力机械子库）

电力机械子库（也称为电机子库）提供了常用的电机模块，包含完整或简化形式的异步电机、同步电机、永磁同步电机、直流电机、激磁系统和水力和蒸汽涡轮和调速系统模型，模块说明见表 8-2-3。电机参数的单位有标幺制和国际单位制两种。电机模块既可以用作电动机，也可以用作发电机。

表 8-2-3　电机子库

模　块　名	用　　途
Asynchronous Machine pu Units	异步电机（标幺制单位）模块
Asynchronous Machine SI Units	异步电机（国际单位）模块
DC Machine	直流电机模型，可用作电动机或发电机
Discrete DC_Machine	离散直流电机
Excitation System	为交流同步机提供励磁控制的模块
Generic Power System Stabilizer	普通电力系统稳定器模块
Hydraulic Turbine and Governor	水轮机和控制器模块，用以和同步发电机配套
Machines Measurement Demux	电机测量单元，将各种电机模型输出的测量信号集分离为单个信号输出
Muti-Band Power System Stabilizer	多频段电力系统稳定器模块
Permanent Magnet Synchronous Machine	交流同步电机，转子为永磁体
Simplified Synchronous Machine pu Units	同步电机简单模块（标幺制单位）
Simplified Synchronous Machine SI Units	同步电机简单模块（国际单位）
Steam Turbine and Governor	汽轮机和控制器模块，用以和同步发电机配套
Synchronous Machine pu Fundamental	同步电机基本模块（标幺制单位）
Synchronous Machine pu Standard	同步电机标准模块（标幺制单位）
Synchronous Machine SI Fundamental	同步电机简单模块（国际单位）

4．Power Electronics（电力电子子库）

电力电子子库提供了 9 种模块，分别是二极管、简化/复杂晶闸管、GTO、理想开关、MOSFET、IGBT、通用桥式电路和三电平桥式电路，模块说明见表 8-2-4。

表 8-2-4　电子电力子库

模　块　名	用　　途
Detailed Thyristor	带 RC 缓冲电路的复杂晶闸管模块
Diode	带 RC 缓冲电路的二极管模块
Gto	带 RC 缓冲电路的 GTO 模块
Ideal Switch	带 RC 缓冲电路的开关模块，开关状态由门极信号控制
IGBT	带 RC 缓冲电路的 IGBT 模块
Mosfet	带 RC 缓冲电路的 Mosfet 模块
Three-Level Bridge	三相桥式整流电路模块
Thyristor	带 RC 缓冲电路的晶闸管模块
Universal Bridge	通用桥模块，可设置为单相或三相桥，可以选择不同的电力电子器件，并且可用以用作整流器或逆变器

5．Measurements（测量子库）

测量子库中的模块有 5 种，分别是电压测量模块、电流测量模块、阻抗测量模块、三相电压电流测量模块和万用表模块，模块说明见表 8-2-5。

表 8-2-5　测量子库

模　块　名	用　　途
Current Measurement	用于检测电流，使用时串联在被测电路中
Impedance Measurement	用于测量一个电路某两点之间的阻抗
Multimeter	多路测量仪，可同时检测系统中多点的多项电量参数
Three-Phase V-I Measurement	可测量三相电路中各相的电压、电流信号，使用时串联在被测电路中
Voltage Measurement	用于检测电压，使用时并联在被测电路中

6．Application Libraries（应用子库）

应用子库中又包含了 3 个子库,分别是分布式电源子库、特种电机子库和 FACTS 子库。分布式电源子库中目前只含有适合于普通风能发电系统的分布式能源模型；特种电机子库中含有特殊的直流/交流电机模块和轴系及减速器模型；FACTS 子库中含有 HVDC 系统模型，基于 FACTS 的电力电子模块和特种变压器。这些模块相对比较专业，可以在具体的应用时参看 Simulink 的帮助。

7．Phasor Elements（相量子库）

包括一个静止无功补偿器模块（Static Var Compensator）。

8．Extra Library（附加子库）

附加子库中包含了上述模块库中没有的其他的电气元件模型，很多是可以用其他器件搭建起来的等价模型，这就简化了电路的连接，涉及控制模块、离散控制模块、离散测量模块、测量模块、相量模块等相关内容，包括 RMS 测量、有效和无功功率计算、傅里叶分析 HVDC 控制、轴系变换、三相 V-I 测量、三相脉冲和信号发生、三相序列分析、三相 PLL 和连续/离散同步 6/12 脉冲发生器等，模块说明见表 8-2-6。

表 8-2-6　附加子库

子　库　名	模　块　名	用　　途
Control Blocks	Synchronized 12-Pulse Generator	12 脉冲逆变器晶闸管同步触发模型
	1-phase PLL	单相锁相环
	1st-Order Filter	一阶滤波器
	2nd- Order Filter	二阶滤波器
	3-phase PLL	三相锁相环
	3-phase Programmable Source	三项可变电源发生器
	Bistable	SR 型双稳态电路模块
	Edge Detector	边缘检测模块
	Monostable	单稳态电路模块
	On/Off Delay	输入信号变化时的延时
	PWM Generator	脉宽调制信号发生器
	Sample & Hold	采样保持模块
	Synchronized 6-Pulse Generator	6 脉冲逆变器晶闸管同步触发模型
	Timer	在指定的时间改变信号

（续表）

子 库 名	模 块 名	用 途
Discrete Measurements	3-phase Instantaneous Active & Reactive Power	三相瞬时有功、无功功率测量仪
	abc_to_dq0 Transformation	将 abc 系统内的信号变换到 dq0 系统中
	Discrete 3-phase PLL-Driven Positive Sequence Active &Reactive Power	离散三相正序有功、无功功率计算
	Discrete 3-phase PLL-Driven Positive Sequence Fundamental Value	离散三相正序基频分量
	Discrete 3-phase Positive Sequence Active &Reactive Power	离散三相正序有功、无功功率
	Discrete 3-phase Positive Sequence Fundamental Value	离散三相正序基频分量
	Discrete 3-phase Sequence Analyzer	离散三相序分量分析仪
	Discrete 3-phase Total Power	离散三相总有功功率
	Discrete Active &Reactive Power	计算有功功率和无功功率
	Discrete Fourier	离散傅里叶变换
	Discrete Mean value	离散均值计算
	Discrete PLL-Driven Fundamental Value	计算输入信号的基频值
	Discrete RMS Value	离散均方根值计算
	Discrete Total Harmonic Distortion	离散总谐波畸变率计算
	Discrete Variable Frequency Mean value	计算输入信号的均值
	dq0-based Active &Reactive Power	dq0 系统中的有功、无功功率测量仪
	dq0_to_abc Transformation	将 dq0 系统内的信号变换到 abc 系统中
	FFT	傅里叶变换
Discrete Control Blocks	Discrete 1-phase PLL	离散的锁相环模型
	Discrete 2nd- Order Filter	离散的二阶滤波器模型
	Discrete 2nd- Order Variable-Tuned Filter	离散的二阶可调谐滤波器
	Discrete 3-phase PLL	离散的三相锁相环
	Discrete 3-phase Programmable Source	离散的三相可变电源发生器
	Discrete 3-phase PWM Generator	离散的三相 PWM 发生器
	Discrete Bistable	离散的 SR 型双稳态电路模型
	Discrete Edge Detector	离散的边缘检测模型
	Discrete Gamma Measurement	离散的对比系数检测器
	Discrete Lead-Lag	离散的超前—滞后模型
	Discrete Monostable	离散的单稳态电路模型
	Discrete On/Off Delay	离散的输入信号延时模型
	Discrete PI Controller	离散的 PI 控制器模型
	Discrete PID Controller	离散的 PID 控制器模型
	Discrete PWM Generator	离散的脉宽调制信号发生器模型
	Discrete Rate Limiter	离散的比率限制器模型
	Discrete Simple & Hold	离散的采样与保持器模型
	Discrete Synchronized 12-Pulse Generator	离散的同步 12 脉冲发生器
	Discrete Synchronized 6-Pulse Generator	离散的同步 6 脉冲发生器
	Discrete Variable Transport Delay	离散的可变传输延时器
	Discrete Virtual PLL	离散的虚拟锁相环模型
	Timer	可编程阶跃信号

（续表）

子 库 名	模 块 名	用 途
Measurements	3-Phase Instantaneous Active & Reactive Power	三项瞬时有功、无功功率测量仪
	3-Phase Sequence Analyzer	输出三相不平衡信号的正、负、零序分量
	abc_to_dq0 Transformation	将 abc 系统内的信号变换到 dq0 系统中
	Active & Reactive Power	有功、无功功率测量仪
	dq0_based Active & Reactive Power	dq0 系统中的有功、无功功率测量仪
	dq0_to_abc Transformation	将 dq0 系统内的信号变换到 abc 系统中
	Fourier	傅里叶变换
	RMS	求均方根值
	Total Harmonic Distorsion	计算总谐波畸变率
Phasor Library	3-Phase Active & Reactive Power (Phasor Type)	相量域的三相有功和无功功率测量仪
	Active & Reactive Power(Phasor Type)	相量域的有功和无功功率测量仪
	Sequence Analyzer(Phasor Type)	相量域的序分析仪
	Static Var Compensator(Phasor Type)	相量域的静止无功补偿器

此外，Simpowersystems 中还含有一个功能强大的图形用户分析工具 Powergui。这些模块可以与标准的 Simulink 模块一起，建立包含电气系统和控制回路的模型，并且可以用附加的测量模块对电路进行信号提取、傅里叶分析和三相序分析。

8.2.3 常用模块设置

MATLAB 中电力电子器件模型使用简化宏模型，只要求外特性与实际基本相符，不考虑内部细微结构。在使用模块之前首先必须熟悉此模块所代表实际电子器件的原理及性能，每个模块都有各自的参数设置，必须详细了解这些参数的物理意义，在此基础上才能正确搭建电路仿真图。

以下给出一些典型模块的功能及参数说明，同类型的模块参数有相似性，可参考典型模块。

1．电源类

模块库中没有直流电流源，可对“AC Current Source”模块进行适当设置，将频率设置为 0。

（1）AC Current Source：交流电流源。

- Peak amplitude（A）：振幅；
- Phase（deg）：初相角；
- Frequency（Hz）：频率（如果为 0 则为直流电流源）；
- Sample time：采样时间，0 为连续来源；
- Measurements：测量（选择电流 Current 则为测量当前电源的电流）。

（2）Controlled Current Source：受控电流源。

- Initialize：设定（如果选择，初始化受控源）；
- Source type：电源类型（可选择 AC 或 DC 受控源）；
- Initial amplitude：初始幅度；
- Initial phase：初始相角；

- Initial frequency：初始频率；
- Measurements：测量。

（3）Three-Phase Programmable Voltage Source：三相可编程电压源。

- Positive-sequence：电源的幅值、初相角、频率；
- Time variation of：时间变化（选择为时间变化编写程序的参数）；
- Type of variation：变化的类型，随着变化类型的选择会出现下列选择参数；
  - ✧ Step magnitude：步长的量级；
  - ✧ Rate of change：变化率；
  - ✧ Amplitude of the modulation：调制的振幅；
  - ✧ Frequency of the modulation：调制的频率；
  - ✧ Variation timing(s)：变化时间安排（s）；
  - ✧ Fundamental and/or Harmonic generation：基波或者谐波的产生。

（4）Three-Phase Source：三相电源。三相电源模块执行带有内部 RL 阻抗的平衡的三相电压来源。各相与中性点连接成 Y 或者其他方式。

- Phase-to-phase rms voltage：线电压有效值；
- Phase angle of phase A：A 相初相角；
- Frequency：频率；
- Internal connection：内部接线方式；
  - ✧ Y：星状接线；
  - ✧ Yn：第四点接地（适合非直接接地系统，可以接入其他元件）；
  - ✧ Yg：中性点直接接地。
- Specify impedance using short-circuit level：根据短路容量确定电抗；
- 3-phase short-circuit level at base voltage：基准电压下的短路容量；
- Base voltage：基准电压；
- X/R ratio：X/R 比值，一般选用短路容量。

2．开关

- Breaker：断路器；
- Breaker resistance Ron：内部电阻；
- Initial state：初始状态设置，0/1 分别表示断路器闭合和断开，如果断路器初始状态被设置为 1（闭合），Simpowersystems 自动地初始化所有线性的电路和断路器初始电流，这样仿真在稳定状态中开始；
- Snubber resistance Rs：过渡电阻，设置为 inf，消除缓冲；
- Snubber capacitance Cs：过渡电容，设置为 0 不考虑过渡电容，设置为 inf 获得一个容抗；
- Switching times：转换时间；
- External control of switching times：转换时间的外部控制；
- Measurements：测量。

3．线路及负载

（1）Distributed Parameter Line：分布参数线路。

- Number of phases N：相数，模块图形随参数变化；
- Frequency used for RLC specifications：LRC 频率，规定被用于计算线路模型的电阻 R，电感 L 和电容 C 矩阵的频率；
- Resistance per unit length：单位长度电阻（Ω/km）；
- Inductance per unit length：单位长度电感（H/km）；
- Capacitance per unit length：单位长度电容（F/km）；
- Line length：线路长度（km）；
- Measurements：测量。

（2）PI Section Line：π 状线路。

- Frequency used for RLC specifications：用于 RLC 的频率；
- Resistance per unit length：单位长度电阻；
- Inductance per unit length：单位长度电感；
- Capacitance per unit length：单位长度电容；
- Length：长度；
- Number of pi sections：型段的数量；
- Measurements：测量。

（3）Series RLC Branch：串联 RLC 支路。

- 模块库中没有单一的电阻、电容或电感，可通过设置 Series RLC Branch 或 Parallel RLC Branch 得到电阻、电容或电感的任意组合电路；
- 纯电阻：参数 Resistance 设置为所仿真电阻的真实值，Inductance 设置为 0，Capacitance 设置为 inf；
- 纯电感：参数 Inductance 设置为所仿真电感的真实值，Resistance 设置为 0，Capacitance 设置为 inf；
- 纯电容：参数 Capacitance 设置为所仿真电感的真实值，Resistance 和 Inductance 均设置为 0。

根据参数设置各类 RC、LC、RLC 等电路。

（4）Parallel RLC Load：并联 RLC 负载。

- Nominal voltage Vn：额定电压；
- Nominal frequency fn：额定频率；
- Active power P：有功；
- Inductive reactive power QL：感性无功；
- Capacitive reactive power QC：容性无功；
- Set the initial capacitor voltage：设置电容初始电压；
- Capacitor initial voltage（V）：电容初始电压值；
- Set the initial inductor current：设置电感初始电流；
- Inductor initial current（A）：电感初始电流值；

- Measurements：测量。

（5）Three-Phase Fault：三相故障。

- Phase A（B）（C）Fault A（B）（C）：相故障，通过这些选项选择故障类型；
- Fault resistances Ron：故障电阻，不能选择 0；
- Ground Fault：接地故障；
- Ground resistance Rg：接地电阻；
- External control of fault timing：外部控制故障时间；
- Transition status：状态转换；
- Transition times(s)：转换时间；
- Snubbers resistance Rp：过渡电阻；
- Snubbers capacitance Cp：过渡电容；
- Measurements：测量。

（6）Three-Phase Dynamic Load：三相动态负载。

可以通过内部时间和外部方式控制有功无功负载。

- Nominal L-L voltage and frequency：额定电压和频率；
- Active and reactive power at initial voltage：初始电压时的有功无功功率；
- Initial positive-sequence voltage Vo：初始正序电压，电压为标幺值，同时输入相角；
- External control of PQ：外部控制 PQ；
- Parameters [np nq]：参数；
- Time constants [Tp1 Tp2 Tq1 Tq2]：连续时间；
- Minimum voltage Vmin：最小电压，设置的动态过程在此值之上不会中断；
- Inputs and Outputs：输入输出，M 输出三个参数，即正序电压 pu、有功功率 P 和无功功率 Q。

（7）Linear Transformer：线性变压器。

模型考虑各侧电阻（R1 R2 R3）和漏电感（L1 L2 L3），还有铁芯的磁化的特点，被一个线性（Rm Lm）的分支模拟。对话框里面的标幺值是以变压器本身容量和电压为基准的。

- Units：单位（选择 pu 为标幺制，SI 为国际单位制）；
- Nominal power and frequency：额定容量和频率；
- Winding 1 2 3　parameters：各侧电压、电阻、电感；
- Three windings transformer：三线圈变压器，选择则显示第三个线圈参数；
- Magnetization resistance and reactance：激磁电阻和电感；
- Measurements：测量。

（8）Saturable Transformer：可饱和变压器。

模型考虑线圈的电阻（R1 R2 R3）和漏感（L1 L2 L3），还有铁芯的磁化的特点，用模拟铁芯的有功损失和一可饱和的电感和电阻模拟。

- Units：单位，可以选择 pu 和 SI；
- Nominal power and frequency：额定容量和频率；

- Winding 1（2）（3）parameters：线圈参数，电压有效值、线圈电阻和漏感；
- Saturation characteristic：饱和特点；
- Core loss resistance and initial flux：铁芯损耗电阻和初始电流；
- Simulate hysteresis：模拟滞后；
- ysteresis data MAT file：滞后 MAT 文件，可以在 Powergui 中 Hysteresis Design tool 设计保存；
- Measurements：测量。

（9）Surge Arrester：过电压保护。

- Protection voltage Vref：保护电压；
- Number of columns：金属氧化物的数量，最小为 1；
- Reference current per column Iref：每个金属氧化物的电流；
- Segment 1（2）（3）characteristics：特性参数。

（10）Three-Phase Harmonic Filter：三相滤波器，使用 RLC 可以组成四种类型的三相滤波器。

- Type of filter：滤波器的类型；
- Filter connection：滤波器连接方式；
- Y(grounded)：中性点接地；
- Y(floating)：中性点不接地；
- Y(neutral)：中性点提供接口；
- Delta：三角形接线；
- Nominal L-L voltage and frequency：额定线电压和频率；
- Nominal reactive power：额定无功功率；
- Tuning frequency or Tuning frequencies：调节频率，单通滤波器的调节频率，或者双调节的滤波器的两种频率。

4．电力电子

Universal Bridge：通用变换器桥，模块由 6 个功率开关元件组成桥式通用三相变换器。功率电子元件的类别和变换器的结构可以通过对话框进行选择。桥的类型有 diode、thyristor、mosfet-diode、igbt-diode、ideal-switch，桥的结构有单相、两相和三相。

- Number of brodge arms：桥臂数量，确定端口形式；
- Snubber resistance Rs：缓冲电阻；
- Snubber capacitance Cs: 缓冲电容；
- Power electronic device：电力电子装置，如选择 THYRISRTOR，即晶闸管型；
- Ron：晶闸管的内电阻；
- Lon：晶闸管的内电感；
- Forward voltage：晶闸管的正向压降；
- Measurements：测量。

5．电机

Simpowersystems 中没有独立区分发电机与电动机，电机模块可工作于电动机或发电机方式。运行方式由电机的电磁转矩符号决定（为正则是电动机状态，为负则是发电机状态）。

- Simplified Synchronous Machine：简化的同步电机；
- Connection type：连接类型；
- Nominal power, line-to-line voltage, and frequency [Pn(VA) Vn(Vrms) fn(Hz)]：额定值，额定功率、线电压、频率；
- Inertia, damping factor and pairs of poles [H(sec) Kd(pu_T/pu_w) p()]：机械特征：惯性因数、阻尼系数和极对数；
- Internal impedance [R(pu) X(pu)]：内部电阻，每相的电阻和电抗值；
- Initial conditions [dw(%) th(deg) ia,ib,ic(A) pha,phb,phc(deg)]：初始状态，初始速率的偏差、转角、线电流幅值和相角。

8.3 电力系统中典型电路的仿真

8.3.1 直流电路仿真

直流电路的数学模型相对简单，仿真大多既可利用编程方式，将电路进行等效转换，列出电路的代数方程进行求解，也可通过 Simulink 建模实现。编程方式可随时任意调整参数，便于分析；而模块方式建模简单直观。

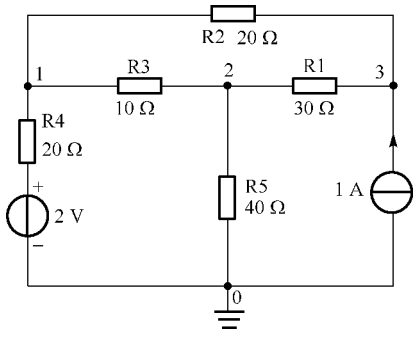


图 8-3-1　例 8-3-1 电路图

在仿真中应注意理想电压源不能与纯电容并联，理想电流源不能与纯电感串联。违反上述原则，电路将无法仿真。方法是在电容旁串联一个小电阻或在纯电感两端并联一个大电阻。

Powergui 模块对于任何包含 Simpowersystems 模块的 Simulink 模型的仿真是必须的，它用于存储等价的 Simulink 电路，这些电路用来表示 Simpowersystem 模块的状态方程。

【例 8-3-1】测量图 8-3-1 所示电路的节点电压。仿真模型如图 8-3-2 所示。

注意事项：

（1）模块库中的直流受控电流源，通过设置“AC Current Source”模块的频率值为 0 实现。

（2）纯电阻模块，可对“Series RLC Branch”模块进行适当设置，将电感 L 设为 0，电容 C 设为 inf。

（3）Simulink 的“Display”模块不能直接接在被测元件两端测量电压，需以“Voltage Measurement”模块作为连接。若显示电流值则通过电流测量模块连接。

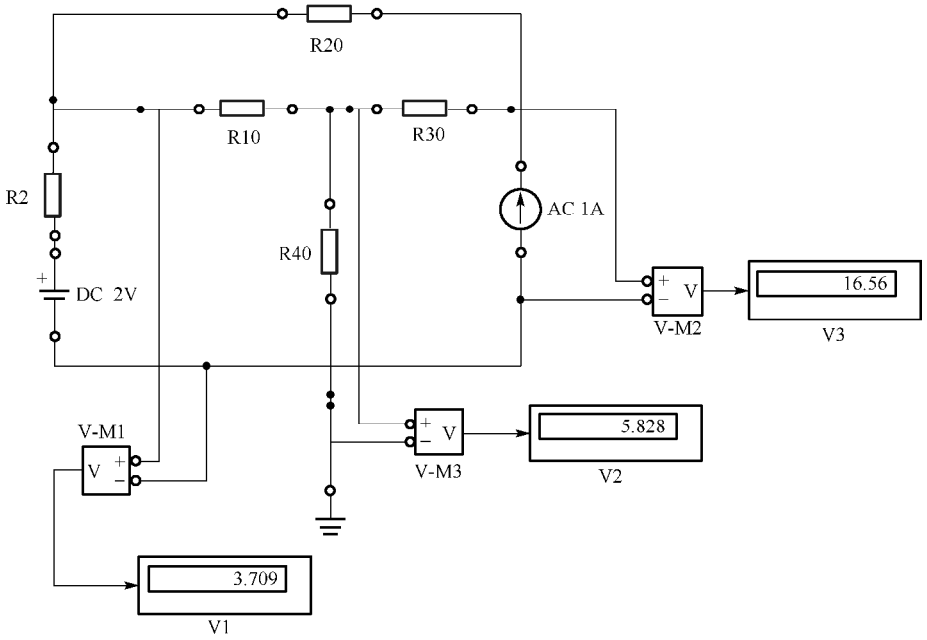


图 8-3-2　直流电路仿真模型

8.3.2　开关电路仿真

模块库中包含两种开关模块：断路器 Breaker 和理想开关 Ideal switch。

断路器实现了一种可以通过外部或内部控制电路开、关时刻的电路。该模型包含了一个串联 RC 缓冲器电路，可以把它连接到电路断路器上。当该模块与电感或电流源串联时，必须在其中加入缓冲器电路。在大多数情况下，可使用纯电感缓冲器（此时将“Snubber capacitance Cs”设为 inf）,而缓冲器电阻设为高阻值的电阻（1e6 等）。由于模型限制，开关电阻不能设为 0。

当断路器处于闭合状态，其等效于一个电阻 R_{on}。R_{on} 可以被设为很小的值（典型值为 10 mΩ），从而与外电阻比较而言可以忽略。当断路器处于断开状态，其等效于一个无穷大的电阻。

理想开关由一个电阻（R_m）、一个电感（L_m）和一个受逻辑信号 G 控制的开关串联构成。电源的断开完全由门极信号（$G>0$ 或 $G=0$）控制。

当 $G=0$，阻断正反向电压及电流为零。

当 $G>0$，电压为零，电流可双向流动。当触发时，开关在开与关状态的转换瞬间完成。

理想开关还包括一个与之并联的 R_s-C_s 串联的缓冲电路。可以指定缓冲电路的 R_s 和 C_s。当指定 $C_s=0$ 时，缓冲电路为纯电阻；当指定 $R_s=0$ 时，缓冲电路为纯电容。

一般来说断路器模块用于交流电路，理想开关用于直流电路。

【例 8-3-2】构建一个包含一个负载，开关，分别由交流电源和直流电源供电的简单回路，观察开关元件的使用方法。

（1）直流电源开关回路。仿真模型如图 8-3-3 所示。

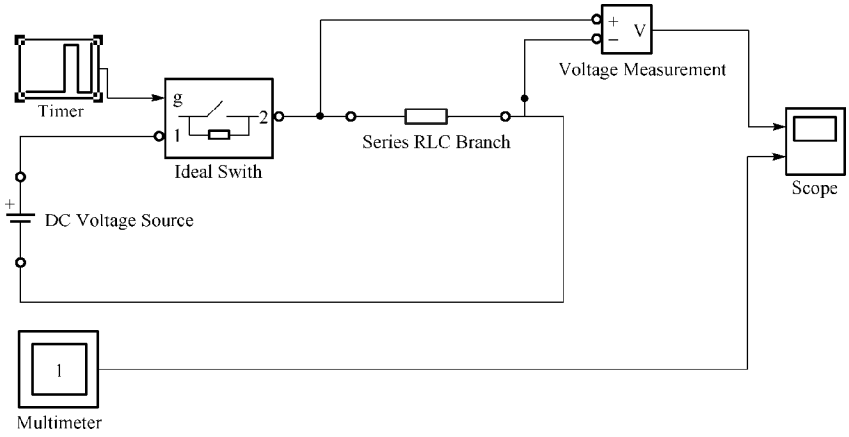


图 8-3-3　理想开关电路仿真模型

Timer 模块的参数设置如图 8-3-4 所示。

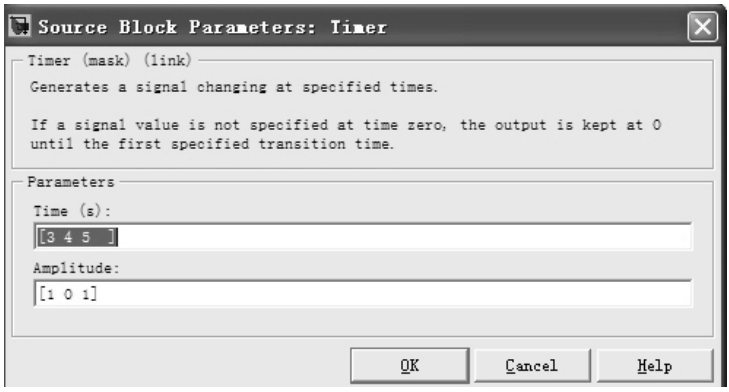


图 8-3-4　Timer 模块的参数设置

输出模型如图 8-3-5 所示。

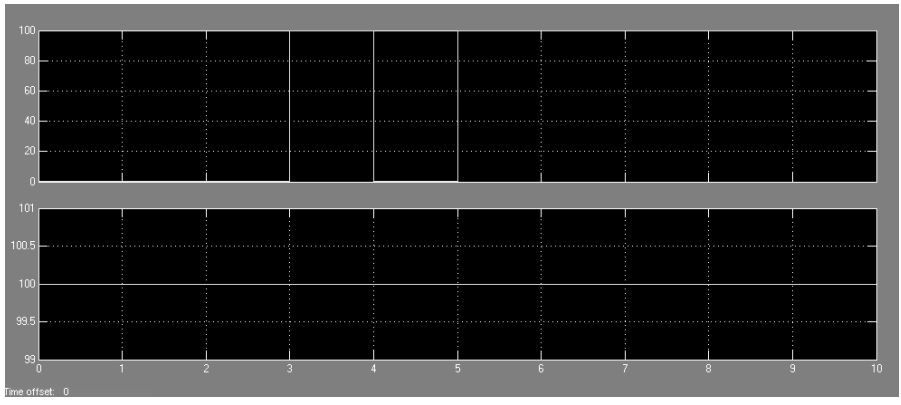


图 8-3-5　输出模型

（2）交流电源开关回路。为便于观察，将交流电频率设为 1。断路器仿真模型如图 8-3-6 所示。

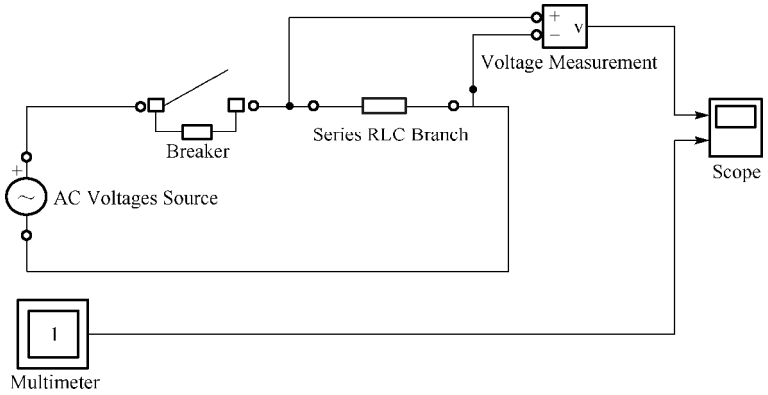


图 8-3-6　断路器仿真模型

Breaker 模块的参数设置如图 8-3-7 所示。

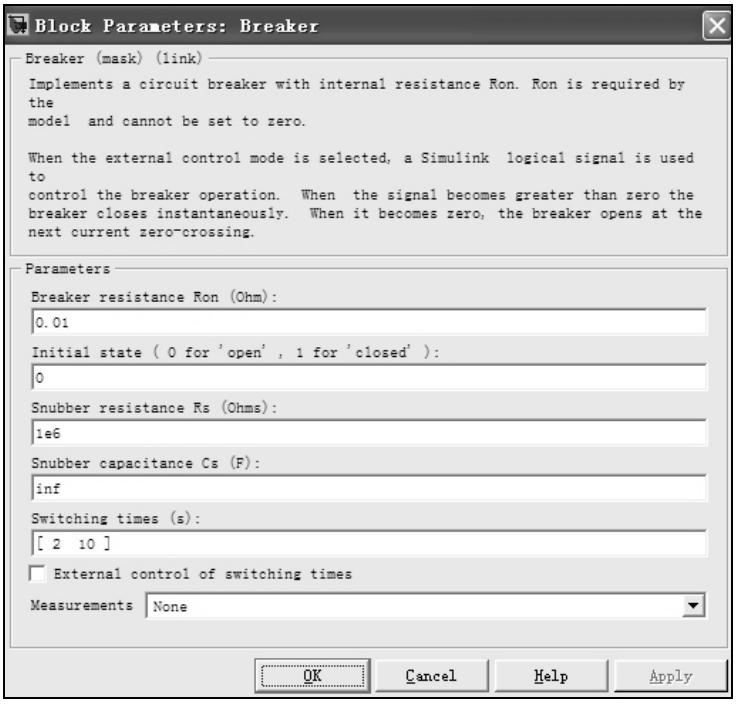


图 8-3-7　Breaker 模块的参数设置

仿真器参数在默认状态下观察的波形如图 8-3-8 所示。

修改仿真器参数观察的波形如图 8-3-9 所示，仿真参数设置窗口如图 8-3-10 所示。

读者可在上述两例图的基础上将 Breaker 模块和 Ideal switch 模块互相代替，或更改控制方式为内部或外部控制，修改 Solver 中的仿真算法和步长参数，会观察到不一样的结果。

仿真分析：

（1）Ideal Switch 一般用于直流电路。Breaker 模块一般用于交流电路，Breaker 的执行能从一个外部 Simulink 信号或者是从一个内部的控制定时器控制一个电路的断开和闭合的状态，但需有过零点，故不适合直流电路。

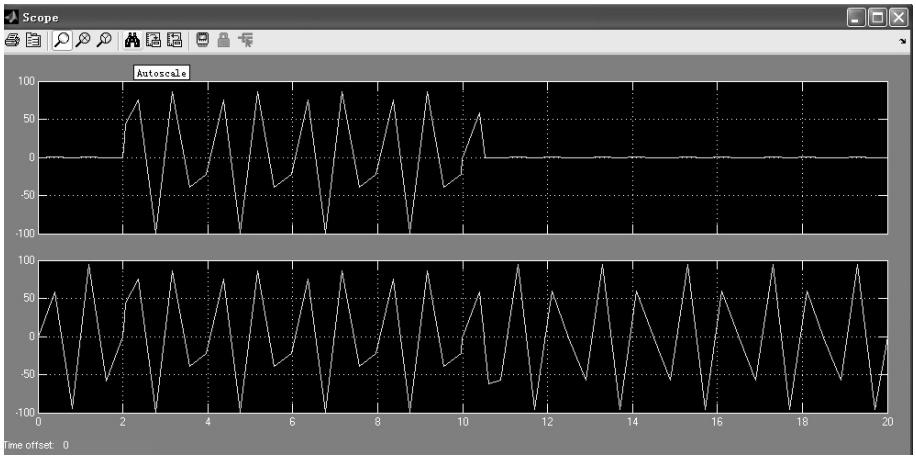


图 8-3-8　仿真器参数在默认状态下观察的波形

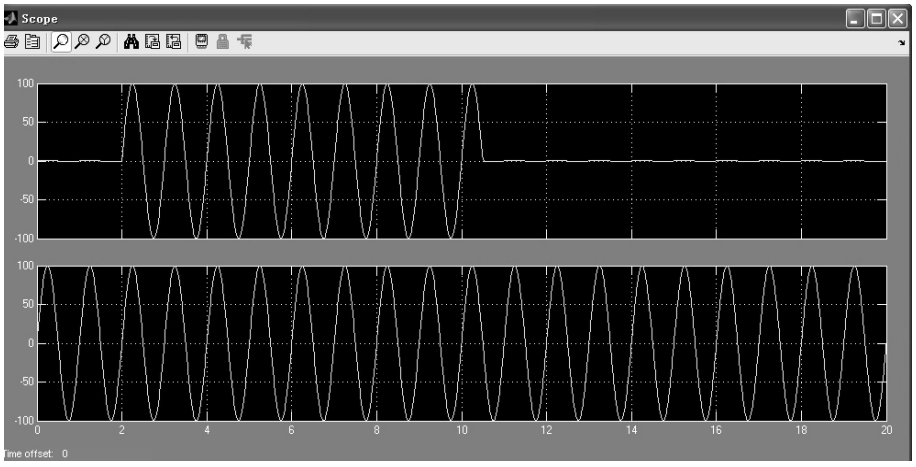


图 8-3-9　修改仿真器参数观察的波形

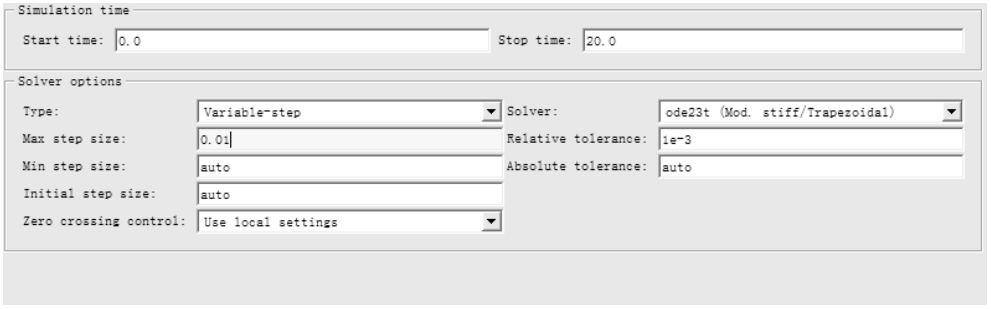


图 8-3-10　仿真参数设置窗口

（2）当与电源连接时，Breaker 模块必须设置过渡电阻、过渡电容，一般将 Rs 设置一个较大的值，Cs 设置为 inf。断路器内部电阻不能设置为 0。

（3）不同的仿真算法和步长，会有很大差别，Breaker 模块必须使用一种刚性综合算法，如 ode23t。最大步长的调整应随系统的具体频率而自行设定。

（4）对于离散化的模型，控制信号保持为 1 的时间应至少为采样周期的 3 倍，否则，电路相当于开路。

8.3.3 整流滤波电路仿真

整流滤波电路可把不稳定的交流变为稳定的直流，三相桥式整流电路是电力电子变流技术中非常重要的一个功能，它不仅可以将交流电压转换成直流电压，以用作直流电动机的直流电源，还可调节电动机电枢电压以进行电动机的调速。在电力电子变流电路中，三相桥式整流电路应用十分广泛。

仿真电路中是将三相交流电源通过三相可控整流桥臂转换成为平均值可以控制改变的直流电源，而平均值的大小改变是通过脉冲触发器控制三组晶闸管的控制角的大小来实现的。同时电路的输出情况与负载的性能有关。

晶闸管整流是电力电子技术中最基础的变流技术，电力电子技术的核心是电力变换也就是变流技术。通过对晶闸管等器件的控制从而实现电力变换。

晶闸管整流桥模块可以直接选用 PowerElectronic 中的 Universal Bridge，定义桥的类型，也可用多组独立的晶闸管模块 Thyristor 连接实现。模块库中，三相电源有独立的三相电源模块，也可由三个单相电源通过设置相位差实现，脉冲触发也有单脉冲模块和 6 脉冲集成模块。整流分为全波和半波两种方式，滤波可分为无源滤波或有源滤波，电路的搭建可形成多种方式。对于初学者来说，建议分别建立各种形式的仿真模型，通过将电压和电流测量模块置于不同的线路点，观察波形，进一步加深对仿真模块特性的认识。

1．采用独立元件构成整流桥

根据原理可以利用 Simulink 内的模块建立仿真模型。设置三个交流电压源 V_a、V_b、V_c 相角依次相差 120°，得到整流桥的三相电源。用 6 个 Thyristor 构成整流桥，实现交流电压到直流电压的转换。6 个pulse convertor 产生整流桥的触发脉冲。从上到下分别给 1 到 6 号晶闸管触发脉冲。

独立元件构成整流桥的仿真模型如图 8-3-11 所示。

参数设置：该模型中电源、晶闸管、触发脉冲均采用独立元件设计。参数设置比较简单，需对每个脉冲发生器（Pulse Generator）设置合理的参数，从而获得三相整流桥所要求的触发脉冲。可设参数为

- 周期（s）：0.02；
- 脉冲占空比：25%；
- 幅值：0.1。

每个脉冲发生器这几项的参数设置均相同，不同之处在于开始时间start time 的设置，根据具体的触发角度，设置每个触发脉冲有一定的相差，实现整流触发。

电路工作正常时，6 个晶闸管的参数可设置为

- 电阻：0.1；
- 电感：10e-6；
- 直流电压源电压：0；

- 初始电流：0；
- 缓冲电阻：100；
- 缓冲电容：0.1e-6。

仿真参数设置：开始时间为 0.04 s（晶闸管第一次触发时间）；停止时间为 0.2 s；运行仿真程序可以得到正常的仿真波形，如图 8-3-12 所示。

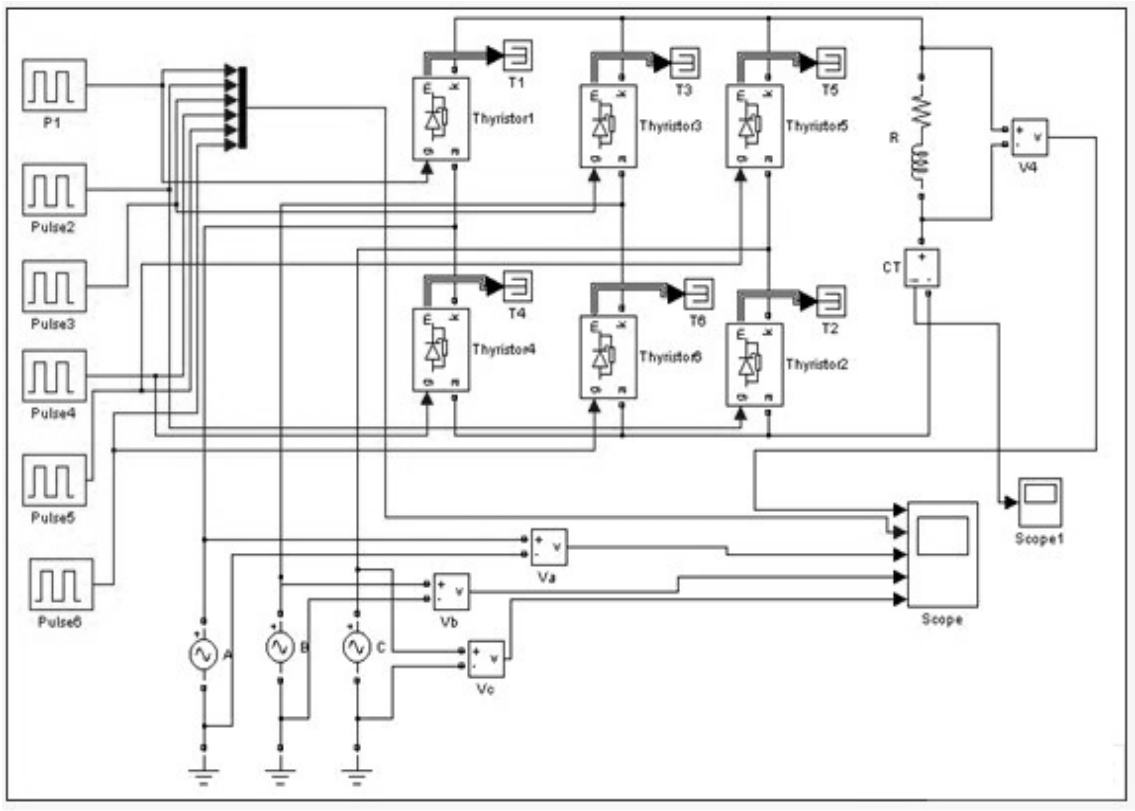


图 8-3-11　独立元件构成整流桥的仿真模型

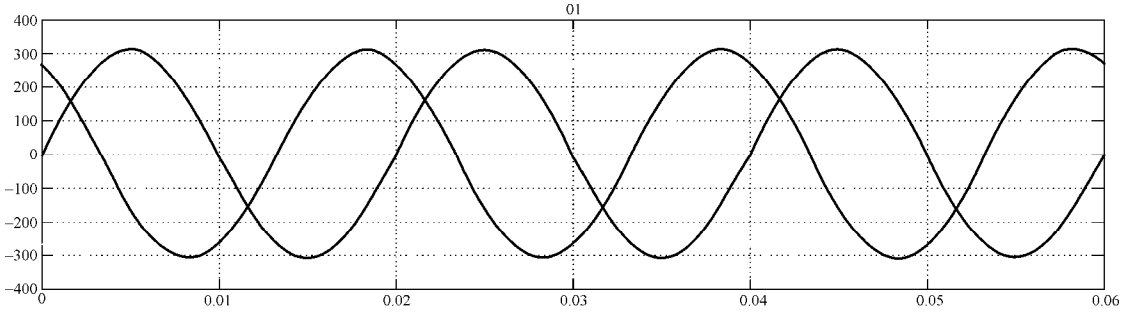


图 8-3-12　输入线电压波形

结果分析：如果是电阻负载，整流后的电压和电流波形相同，但幅值不同。改变控制角可观察在不同控制角下整流器的工作情况，如图 8-3-13 所示。

整流器二次侧各相电流波形反映了晶闸管中流过电流的波形，从波形可以看出，晶闸

管一周期中有 120° 处于通态，240° 处于断态，由于负载为电阻，故晶闸管处于通态时的电流波形与相应时段的电压波形相同。可以修改负载特性观察波形，如纯电阻负载，RL 负载等；也可以利用触发脉冲参数的改变，仿真不同负载与不同触发角情况下的波形。

晶闸管出现故障的几率较大，故障情况分几种，如同相中晶闸管故障，不同相中的故障等，通过设置故障可观察晶闸管故障时的波形，仿真波形在此不一一列举。

2．采用整流桥模块

电路原理与上述相似，电源中利用变压器模块进行降压处理，直接利用模块库中的整流桥模块实现整流控制。脉冲发生器选用 6 脉冲发生器。采用晶闸管整流桥模块的仿真模型如图 8-3-14 所示。

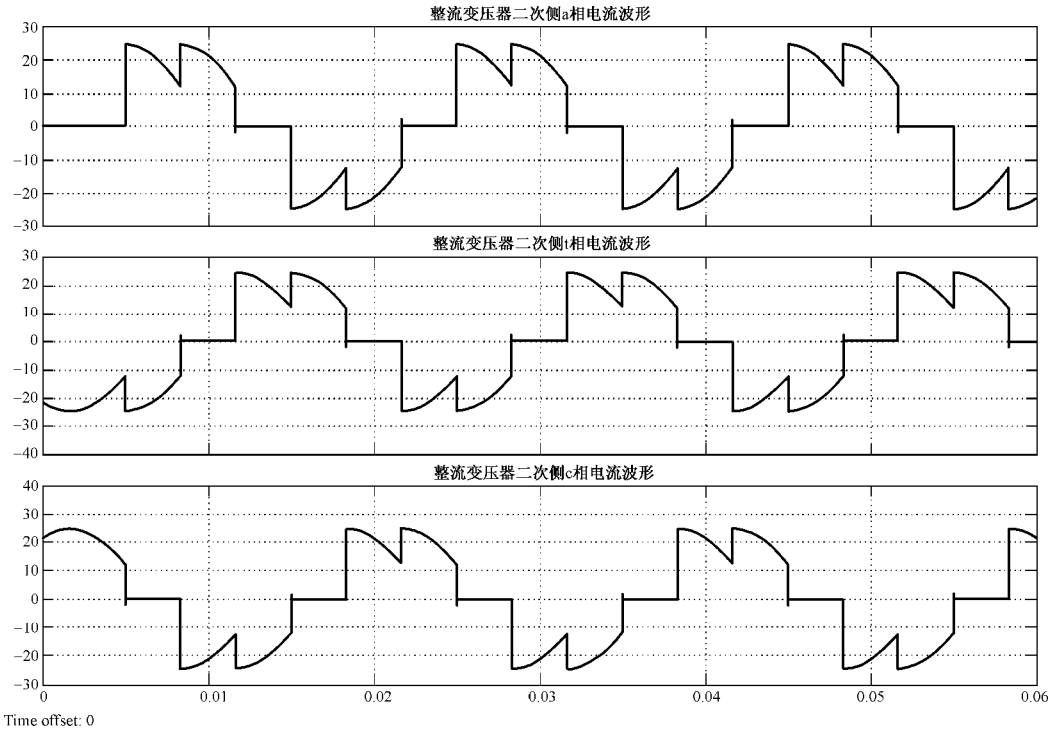


图 8-3-13　整流器二次侧各相电流波形

参数设置：

（1）三相交流电压源：分别将 ua、ub、uc 参数中的 Phase（deg）（相位）修改为 0、–120、–240；峰值（Peak amplitude）和频率（Frequency）分别改为 311、50。

（2）万用表（Multimeter）：选择 RL 两端的电压、电流；负载 Rl 为纯电阻电路：RL=5

（3）同步脉冲触发器：脉冲触发器用于触发三相全控整流桥的 6 个晶闸管，同步 6 脉冲触发器可以给双脉冲，双脉冲间隔为 60°。触发器输出的 1～6 号脉冲依次送给三相全控整流桥对应标号的 6 个晶闸管。同步脉冲触发器包括同步电源和 6 脉冲触发器两个部分。6 脉冲同步触发器包括 5 个输入端和 1 个输出端，各部分功能如下所述。

- Alpha_deg：此端子为移相控制角信号的输入端，单位是度；

- AB、BC、CA：三相电源的三线电压输入，同步线电压就是整流桥的三相交流电压的线电压；
- BLOCK：触发器控制端，输入为 0 时，开放触发器，输入大于 0 时，则锁存触发器。故用于与触发器模块的开通与封锁操作；
- PLUSE：6 脉冲输出信号。

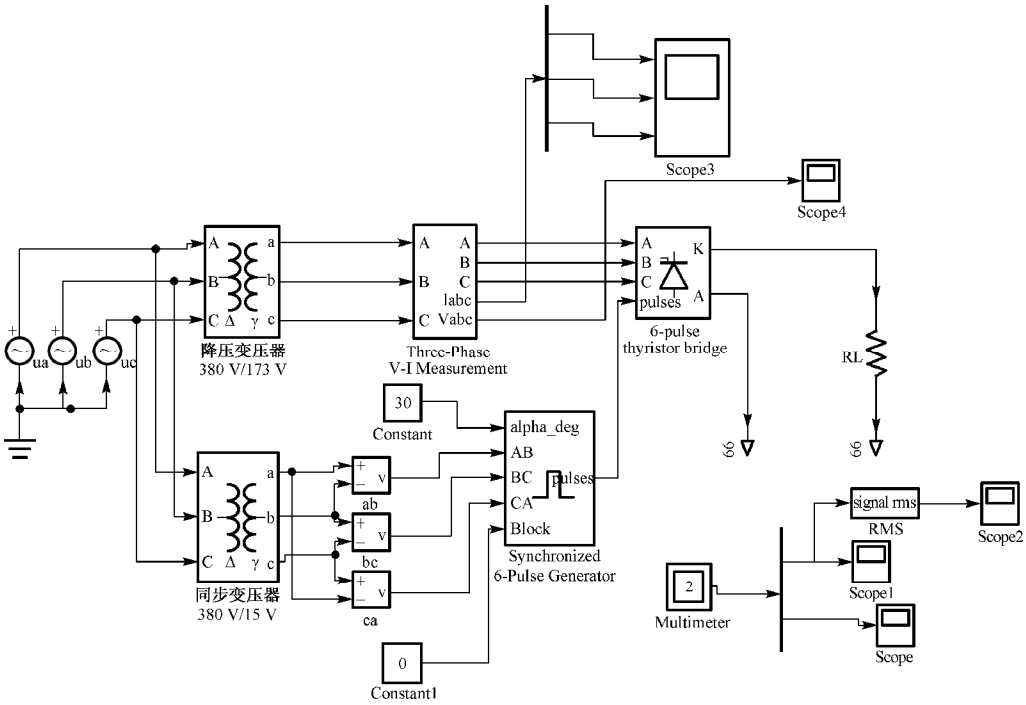


图 8-3-14　采用晶闸管整流桥模块的仿真模型

主要模块参数设置如图 8-3-15、图 8-3-16、图 8-3-17 和图 8-3-18 所示。

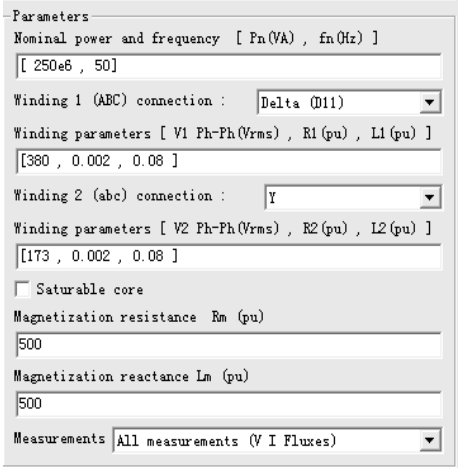


图 8-3-15　降压变压器的参数修改

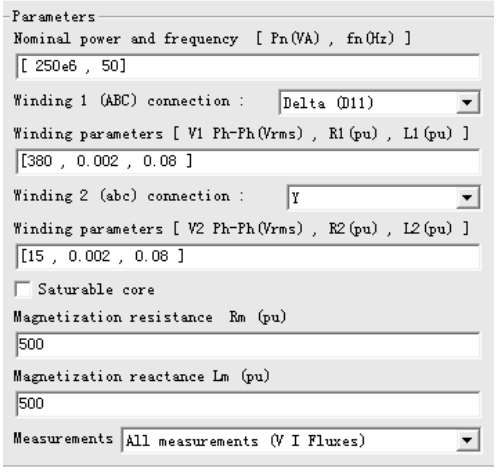


图 8-3-16　同步变压器的参数修改

仿真并观察结果。设置仿真时间 0.06 s，数值算法采用 ode15。

当触发角 *a* 改变时，电路的工作情况也将会发生变化。

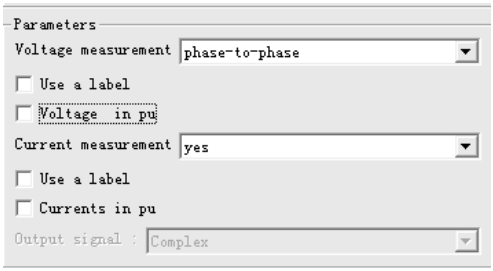


图 8-3-17　V-I 测量器的参数修改

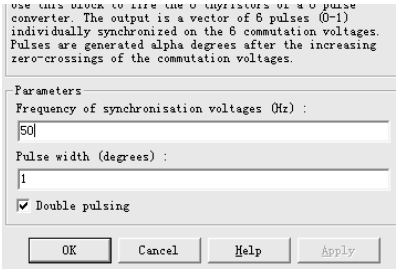


图 8-3-18　同步六脉冲触发器的参数修改

3．整流滤波的谐波分析

整流滤波后可以通过加入简化模型描述的直流电机负载及 Powergui/FFT 工具描述三相六脉冲整流滤波器的工作状态及负载端的谐波等性能。仿真模型见图 8-3-19。

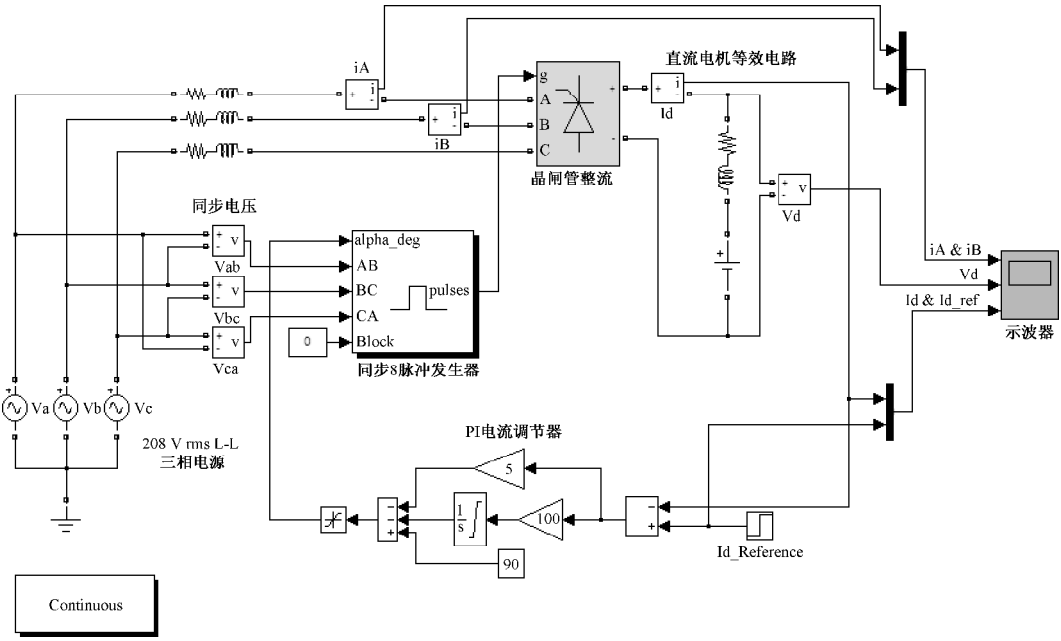


图 8-3-19　仿真模型

整流器的输出电流由 PI 电流调节器调节，阶跃信号用于测试电流调节器的动态性能，负载端的电流及电压波形见图 8-3-20。

为了进一步的信号处理，将信号显示输出存储在一个变量供 Powergui 调用。

仿真完成后，打开 Powergui 并选择 FFT 分析显示 0～2 000 Hz 频率谱信号。FFT 窗口窗口显示从 $t = 0.1–2/60$ 的两个周期的信号及谐波图形，见图 8-3-21。

系统各模块的参数设置，系统解法器中仿真时间、步长、算法等参数的设置都直接影响仿真输出波形。读者可根据实际自己调节参数及感兴趣的变量波形。

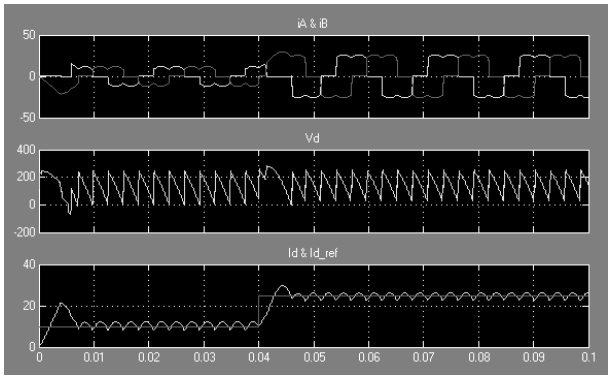


图 8-3-20　输出电流及电压波形

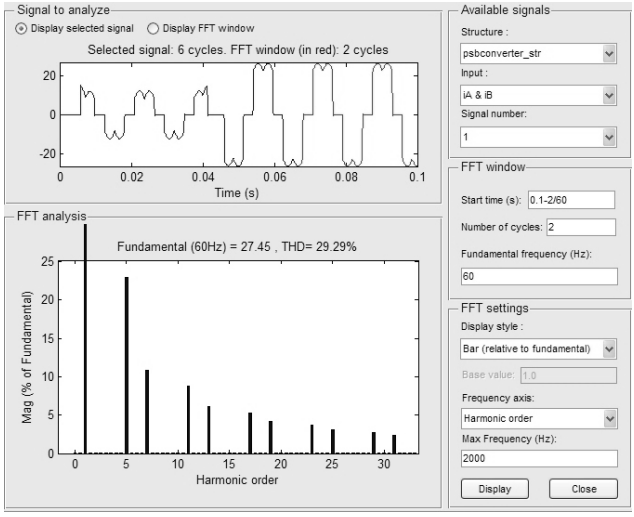


图 8-3-21　输出信号及谐波频谱分析

8.3.4　电力传输系统仿真

1．输电线路模型

电力传输系统中输电线路模型的选择应根据线路的特性选择不同的模块。

对于电压等级不高的短线路，通常忽略线路电容影响，用 RLC 串联支路来等效。通过修改 RLC 串联支路模型参数，建立 R、L、C 三个独立元件及其任意组合的电路模型。

长度大于 100 km 的架空线路及较长的电缆线路，电容影响不能忽略，在进行潮流计算、暂态稳定性分析时常用 PI 型等效电路。

长度不超过 300 km 的线路可用一个 PI 型电路来代替，对于更长的线路，可用串级连接的多个 PI 型电路来模拟。PI 型电路限制了线路中电压、电流的频率变化范围，对于研究基频下的电力系统以及电力系统与控制系统之间的相互关系，PI 型电路可达到足够的精度。

在分析线路的瞬变过程、线路的波过程以及进行更精确的分析时，需要使用分布参数模块，可通过参数设置修改线路的相数。

2．负荷模型

电力系统的负荷相当复杂，不但数量大、分布广、种类多，而且其工作状态带有很大的随机性和时变性，连接各类用电设备的配电网结构也可能发生变化。因此，如何建立一个既准确又实用的负荷模型，至今仍是一个尚未得到很好解决的问题。

通常负荷模型分为静态模型和动态模型，其中静态模型表示稳态下负荷功率与电压和频率的关系；动态模型反映电压和频率急剧变化时负荷功率随时间的变化。常用的负荷等效电路有含源等效阻抗支路、恒定阻抗支路和异步电动机等效电路。

负荷模型的选择对分析电力系统动态过程和稳定问题都有很大的影响。在潮流计算中，负荷常用恒定功率表示，必要时也可以采用线性化的静态特性。在短路计算中，负荷可表示为含源阻抗支路或恒定阻抗支路；在稳定计算中，综合负荷可表示为恒定阻抗或不同比例的恒定阻抗和异步电动机的组合。

Simpowersystems 库中提供了四种静态负荷模块，分别为单相串联 RLC 负荷（Series RLC Load）、单相并联 RLC 负荷（Parallel RLC Load）、三相串联 RLC 负荷（Three-Phase Series RLC Load）和三相并联 RLC 负荷（Three-Phase Parallel RLC Load），单相串联和并联 RLC 负荷模块分别对串联和并联的线性 RLC 负荷进行模拟。在指定的频率下，负荷阻抗为常数，负荷吸收的有功和无功功率与电压的平方成正比。

Simpowersystems 库中提供的三相动态负荷（Three-Phase Dynamic Load）模块是对三相动态负荷的建模，其中有功和无功功率可以表示为正序电压的函数或者直接受外部信号的控制。由于不考虑负序和零序电流，因此即使在负荷电压不平衡的条件下，三相负荷电流仍然是平衡的。

Simpowersystems 中异步电动机模块用四阶状态方程描述电动机的电气部分。

3．高压直流输电系统（HVDC）的仿真

HVDC 是一项极具吸引力的技术，因为在输电过程中，HVDC 技术的电能损耗低于传统交流输电技术的损耗，同时 HVDC 需要的传输线缆更少，能减少占地。由于交流和直流电间的转化需要特殊的设备，因此 HVDC 一般在远距离输电时才体现出经济效益。例如，单极大地回流直流输电系统的基本结构如图 8-3-22 所示。

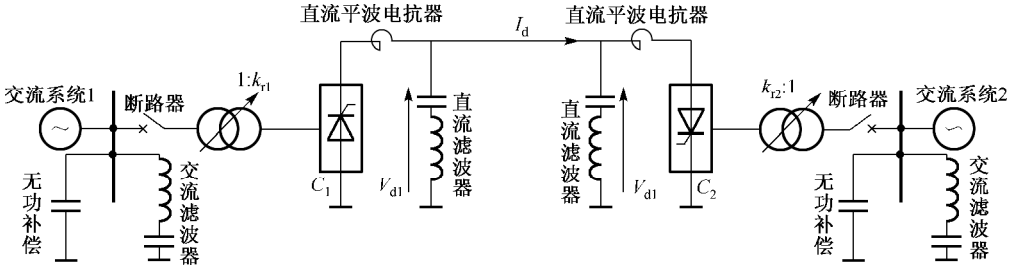


图 8-3-22　单极大地回流直流输电系统的基本结构

1）HVDC 系统的基本结构与工作原理

- 换流变压器：其一次绕组与交流电力系统相连，其作用是将交流电压变为桥阀所需电压。

- 换流器 C1、C2：由晶闸管组成，用做整流和逆变，实现交流电与直流电之间的转换。
- 滤波器：交流侧滤波器一般装在换流变压器的交流侧母线上。
- 无功补偿装置：换流器在运行时需要从交流系统吸引大量无功功率，在稳态时吸收的无功功率约为直流线路输送有功功率的50%，因此，在换流器附近应有无功补偿装置为其提供无功电源。
- 直流平波电抗器：减小直流电压、电流的波动，受扰时抑制直流电流的上升速度。

2）HVDC 系统的仿真模型

HVDC 系统的仿真模型如图 8-3-23 所示，整流环节子系统结构如图 8-3-24 所示，整流器子系统结构如图 8-3-25 所示，滤波器子系统结构如图 8-3-26 所示，整流器控制和保护子系统包含的模块及作用见表 8-3-1，逆变器控制和保护子系统包含的模块及作用见表 8-3-2。

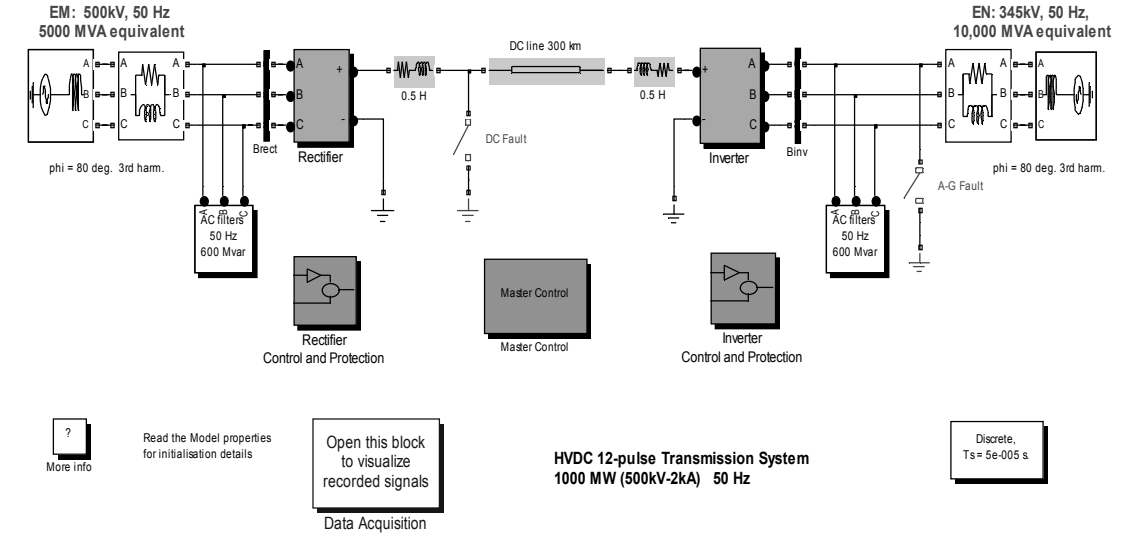


图 8-3-23　HVDC 系统的仿真模型

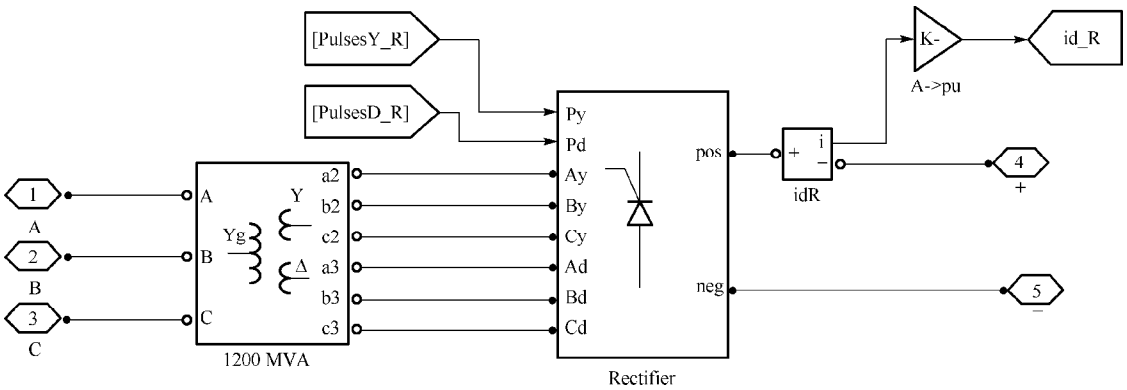


图 8-3-24　整流环节子系统结构

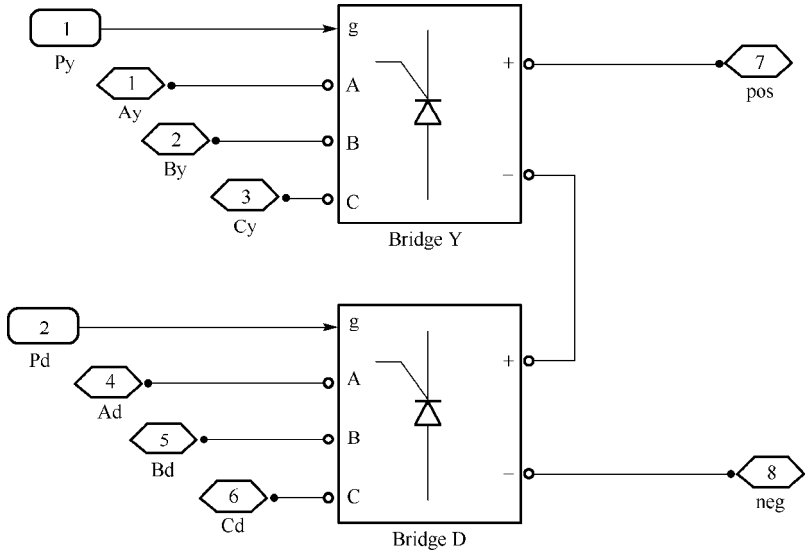


图 8-3-25　整流器子系统结构

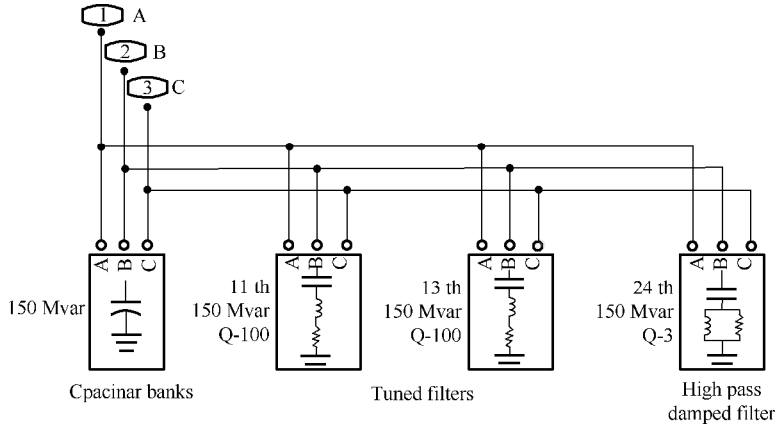


图 8-3-26　滤波器子系统结构

表 8-3-1　整流器控制和保护子系统包含的模块及作用

模 块 名 称		作　　用
Rectifier Controller	Voltage Regulator	电压调节，计算触发角
	Gamma Regulator	计算熄弧角
	Current Regulator	电流调节，计算触发角
	Voltage Dependent Current Order Limiter	根据直流电压值改变参考电流值
Rectifier　Protections	Low AC Voltage Detection	直流侧故障和交流侧故障检测
	DC Fault Protection	判断直流侧是否发生故障，启动必要的动作清除故障
12 Pulse Firing Control		产生同步的 12 个触发脉冲

表 8-3-2　逆变器控制和保护子系统包含的模块及作用

模 块 名 称		作　　用
Inverter Current/ Voltage/Gamma Controller		逆变侧电压、电流、熄弧角调节，与整流侧系统相同
Inverter Protection	Low AC Voltage Detection	交流侧故障检测
	Commutation Failure Prevention Control	减弱电压跌落导致的换相失败
12 Pulse Firing Control		产生同步的 12 个触发脉冲
Gamma Measurement		熄弧角测量

3）HVDC 系统的起停和阶跃响应仿真

晶闸管在 0.02 s 时导通，电流开始增大，在 0.3 s 时达到最小稳态参考值 0.1 p.u.，同时直流线路开始充电，使得直流电压为 1.0 p.u.，整流器和逆变器均为电流控制状态。

在 0.4 s 时，参考电流从 0.1 p.u.斜线上升到 1.0 p.u.（2 kA），0.58 s 时直流电流到达稳定值，整流器为电流控制状态，逆变器为电压控制状态，直流侧电压维持在 1 p.u.（500 kV。

在 0.7 s 时，参考电流出现–0.2 p.u.的变化，在 0.8 s 时恢复到设定值。

在 1.0 s 时，参考电压出现–0.1 p.u.的偏移，在 1.1 s 时恢复到设定值。

在 1.4 s 时，触发信号关断，使得电流斜线下降到 0.1 p.u.。

在 1.6 s 时，整流器侧的触发延迟角被强制设置为 166°，逆变器侧的触发延迟角被强制设置为 92°，使得直流线路放电。

在 1.7 s 时两个变换器均关断，变换器控制状态为 0。

系统控制参数随时间变化见表 8-3-3，变换器控制状态及意义见表 8-3-4。

表 8-3-3　系统控制参数随时间变化表

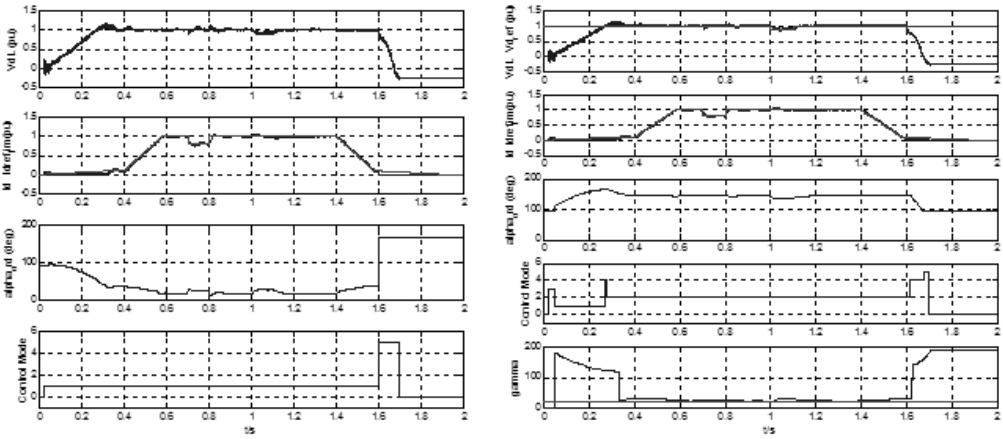
序　　号	时刻/s	动　　作
1	0	电压参考值为 1 p.u.
2	0.02	变换器导通，电流增大到最小稳态电流参考值
3	0.4	电流按指定的斜率增大到设定值
4	0.7	参考电流值下降 0.2 p.u.
5	0.8	参考电流值恢复到设定值
6	1.0	参考电压由 1 p.u.跌落到 0.9 p.u.
7	1.1	参考电压恢复到 1 p.u.
8	1.4	变换器关断
9	1.6	强迫设置触发延迟角到指定值
10	1.7	关断变换器

表 8-3-4　变换器控制状态及意义

状　　态	意　　义	状　　态	意　　义
0	关断	4	α最大值限制
1	电流控制	5	α的设定值或者常数
2	电压控制	6	γ控制
3	α最小值限制		

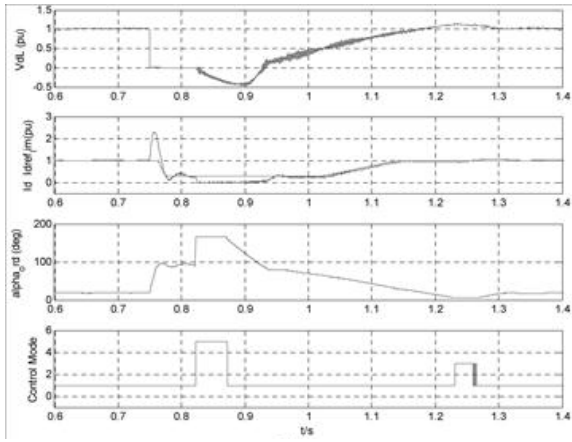
HVDC 系统的起停和阶跃响应仿真波形如图 8-3-27 所示，HVDC 系统直流线路故障仿

真波形如图 8-3-28 所示，HVDC 系统交流侧线路故障仿真波形如图 8-3-29 所示，逆变器交流侧线路故障时三相电压和电流波形如图 8-3-30 所示。

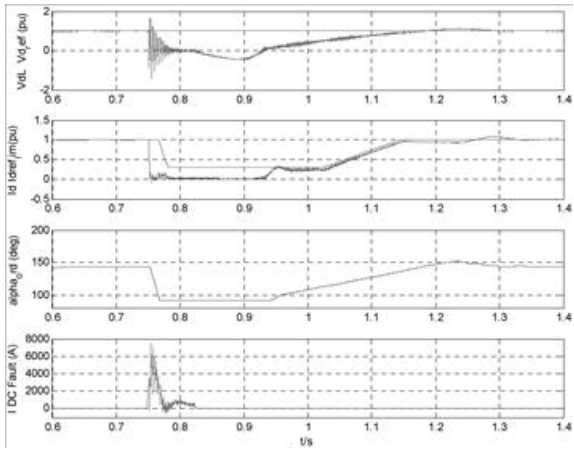


（a）整流侧得到的相关波形　　（b）逆变侧得到的相关波形

图 8-3-27　HVDC 系统的起停和阶跃响应仿真波形图

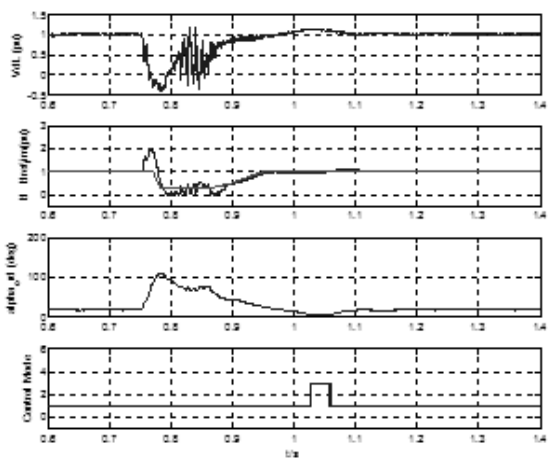


（a）整流侧得到的相关波形

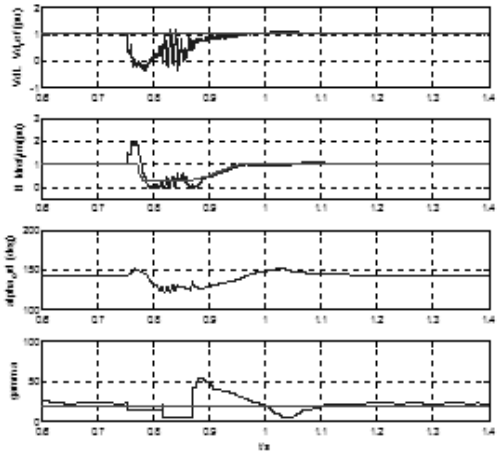


（b）逆变侧得到的相关波形

图 8-3-28　HVDC 系统直流线路故障仿真波形图



（a）整流侧得到的相关波形



（b）逆变侧得到的相关波形

图 8-3-29　HVDC 系统交流侧线路故障仿真波形图

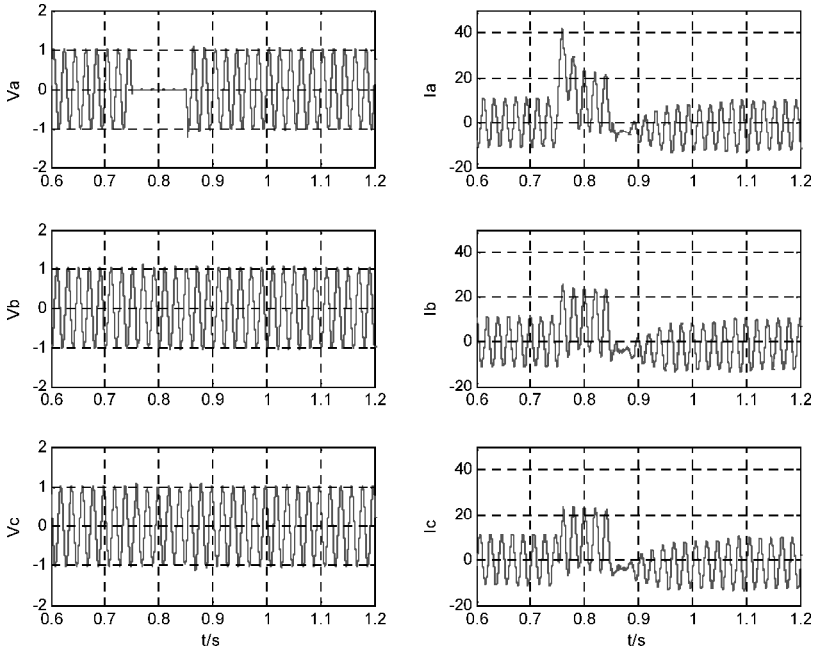


图 8-3-30　逆变器交流侧线路故障时三相电压和电流波形图

参 考 文 献

[1] 瞿亮．基于 MATLAB 的控制系统计算机仿真．北京：清华大学出版社，2006．

[2] 黄忠霖．控制系统 MATLAB 计算及仿真．北京：国防工业出版社，2001．

[3] 薛定宇．控制系统计算机辅助设计——MATLAB 语言及应用．北京：清华大学出版社，2000．

[4] 韩利竹，王华．MATLAB 电子仿真与应用．北京：国防工业出版社，2003．

[5] 孙亮．MATLAB 语言与控制系统仿真．北京：北京工业大学出版让，2001．

[6] 张晓华．控制系统数字仿真与 CAD（第 3 版）．北京：机械工业出版社，2010．

[7] 张晋格．控制系统 CAD——基于 MATLAB 语言．北京：机械工业出版社，2011．

[8] 王晶，张有兵．电力系统的 MATLAB/Simulink 仿真与应用．西安：西安电子科技大学出版社，2007．

[9] 姚俊，马松辉．Simulink 建模与仿真．西安：西安电子科技大学出版社，2002．

[10] 杨立．计算机控制与仿真技术．北京：中国水利水电出版杜，2003．

[11] 林健．MATLAB 在图像处理与目标识别方面的应用．北京理工大学计算机科学技术学院，2010．

[12] 邱晓林，等．基于 MATLAB 的动态模型与系统仿真工具——Simulink 3.0/4.0．西安：西安交通大学出版社，2003．